

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

**Planificador personal adaptado a personas con diversidad
funcional intelectual**

Pablo Ángel Hernández Pallejá
Tutor: Juan Carlos Torrado
Ponente: Germán Montoro Manrique

MAYO 2018

Planificador personal adaptado a personas con diversidad funcional intelectual

AUTOR: Pablo Ángel Hernández Pallejá

TUTOR: Juan Carlos Torrado

PONENTE: Germán Montoro Manrique

**Dpto. Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
mayo de 2018**

Resumen (castellano)

Este Trabajo Fin de Grado se aplica para el diseño y desarrollo de una aplicación, para dispositivos Android, con la que poder mejorar y facilitar la vida de personas con diversidad funcional intelectual, con trastornos como el TEA. La aplicación sirve como una herramienta para la planificación y organización de las tareas diarias que puedan surgir en la vida de los distintos usuarios. Para la planificación de las tareas se facilita una interfaz, intuitiva y adaptada mediante pictogramas, por donde se pueda navegar de forma simple y crear tareas como el usuario desee.

En esta aplicación se pueden crear tareas que el usuario tiene que realizar diariamente o en una fecha en concreto, para la construcción de las tareas se han incluido cuatro principales partes que contiene una tarea para su correcta realización. Estos apartados son los objetos que se debe llevar para poder realizar la tarea, las diferentes situaciones que nos rodean socialmente y que hay que tener en cuenta para realizar la tarea con normalidad, el tiempo de duración de cada tarea y los pasos a seguir para terminar la tarea exitosamente.

El usuario final de la aplicación son las personas que padecen TEA, es por ello que el enfoque de la aplicación esta realizado con la idea de que profesores o tutores de este tipo de personas puedan programar las tareas, de sus alumnos o tutelados, en sus dispositivos o para que los usuarios que tengan las capacidades de realizar la programación de sus propias tareas puedan utilizarla.

Dentro de la aplicación se organizan las tareas de dos maneras distintas, una según la fecha en la que se debe realizar que estarán incluidas en la sección llamada Agenda y otra según a la categoría de tareas a la que pertenezcan que estarán en la sección Categorías.

Palabras clave (castellano)

TEA, Tarea, Categoría, Objeto, Sociedad, Paso, Pictograma, Araasac, Agenda.

Abstract (English)

This bachelor is applied to the design and development of an application for Android devices, to improve and facilitate the lives of people with intellectual functional diversity, with disorders such as ASD. The application serves as a tool for planning and organizing daily tasks that may arise in the lives of different users. For the planning of the tasks an interface is provided, intuitive and adapted by pictograms, where you can navigate in a simple way and create tasks as the user wishes.

In this application you can create tasks that the user has to perform daily or on a specific date, for the construction of the tasks four main parts have been included that contains a task for its correct execution. These sections are the objects that must be carried to perform the task, the different situations that surround us socially and that must be taken into account to perform the task normally, the duration of each task and the steps to follow to complete the task successfully.

The final user of the application is the people who suffer from ASD, that is why the focus of the application is made with the idea that teachers or tutors of this type of people can program the tasks, of their students or mentees, in their devices or so that users who have the capabilities to perform the programming of their own tasks can use it.

Within the application tasks are organized in two different ways, one according to the date in which it must be done that will be included in the section called Agenda and another according to the category of tasks to which they belong that will be in the Categories section.

Keywords (English)

TEA, Task, Category, Object, Society, Step, Pictogram, Araasac.

Agradecimientos

Primeramente, quiero agradecer a mis padres y a mi hermano por estar ahí siempre para escucharme, apoyarme y motivarme para cumplir mis objetivos.

A mi novia, por su apoyo, por la felicidad que me aporta, por la energía y fuerza que me transmite y por todos esos momentos compartidos.

A mi familia y a esos amigos que considero parte de mi familia, por ayudarme a desconectar en momentos necesarios para seguir con más fuerza estudiando.

A todos esos profesores de la carrera que se preocupan por sus alumnos y les apoyan para aprender y superar sus retos.

A mi tutor del TFG Juan Carlos Torrado y a Javier Gómez por todas esas reuniones en este último año donde han compartido conmigo todo lo que saben y me ha servido de gran ayuda.

INDICE DE CONTENIDOS

1	Introducción.....	1
1.1	Motivación	1
1.2	Objetivos	1
1.3	Organización de la memoria	2
2	Estado del arte	3
2.1	Introducción	3
2.2	Día a día (organizador)[6]	3
2.3	TimeTune – Optimiza Tu Tiempo[7]	3
2.4	PictogramAgenda[8]	3
2.5	Planificador Semanal[9]	3
2.6	AssisT-Task[10]	4
2.7	Conclusiones	4
3	Metodología y análisis de requisitos	5
3.1	Modelo de proceso	5
3.2	Diagrama de Gantt	6
3.3	Análisis de requisitos funcionales	8
3.4	Análisis de requisitos no funcionales	10
4	Diseño.....	11
4.1	Casos de uso.....	11
4.2	Modelo-Vista-Controlador (MVC)	13
4.1	Diagrama de clases.....	13
4.2	Maquetas	15
5	Desarrollo	23
5.1	Diagrama de bases de datos	24
5.2	Código del modelo	25
5.3	Código de la vista.....	26
5.4	Código controlador gesto swipe.....	28
5.5	Código selección pictogramas Araasac.....	29
5.6	Pantallazos aplicación	30
6	Pruebas y resultados	31
7	Conclusiones y trabajo futuro.....	32
7.1	Conclusiones	32
7.2	Trabajo futuro.....	32
	Referencias	33
	Glosario	34
	Anexos	- 1 -
A	Casos de uso.....	- 1 -
B	Pantallazos aplicación	- 9 -

INDICE DE FIGURAS

ILUSTRACIÓN 2-1: DÍA A DÍA	3
ILUSTRACIÓN 2-2: TIME TUNE	3
ILUSTRACIÓN 2-3: PICTOGRAMA AGENDA	3
ILUSTRACIÓN 2-4: PLANIFICADOR SEMANAL	3
ILUSTRACIÓN 2-5: ASSIST-TASK	4
ILUSTRACIÓN 3-1: MODELO DE PROCESO DE MEJORA ITERATIVA EN CASCA	5
ILUSTRACIÓN 3-2 DIAGRAMA DE GANTT	7
ILUSTRACIÓN 4-1 DIAGRAMA DE CASOS DE USO	11
ILUSTRACIÓN 4-2 MODELO-VISTA-CONTROLADOR	13
ILUSTRACIÓN 4-3 DIAGRAMA DE CLASES DEL MODELO	14
ILUSTRACIÓN 4-4 MAQUETA SPLASHVIEW	15
ILUSTRACIÓN 4-5 MAQUETA PANTALLA PRINCIPAL	16
ILUSTRACIÓN 4-6 LISTA TAREAS MODO AGENDA	16
ILUSTRACIÓN 4-7 VISUALIZACIÓN DE TAREAS POR DÍA EN MODO AGENDA	17
ILUSTRACIÓN 4-8 CATEGORÍAS	17
ILUSTRACIÓN 4-9 LISTA DE TAREAS DENTRO DE UNA CATEGORÍA	18
ILUSTRACIÓN 4-10 MODO EDICIÓN DE CATEGORÍAS	18
ILUSTRACIÓN 4-11 PASO 1 DE TAREA	19
ILUSTRACIÓN 4-12 PASO 2 DE TAREA	19
ILUSTRACIÓN 4-13 PASO 3 DE TAREA	20
ILUSTRACIÓN 4-14 NOTIFICACIÓN DE FALTA DE OBJETOS	20
ILUSTRACIÓN 4-15 PASO 1 DE EDITAR TAREA	21
ILUSTRACIÓN 4-16 PASO 2 DE EDITAR TAREA	21
ILUSTRACIÓN 4-17 PASO 3 DE EDITAR TAREA	22
ILUSTRACIÓN 4-18 SELECCIÓN DE IMÁGENES	22
ILUSTRACIÓN 5-1 DIAGRAMA BASE DE DATOS	24
ILUSTRACIÓN 5-2 CÓDIGO PARA CARGAR Y ELIMINAR IMAGEN DE DISPOSITIVO	25
ILUSTRACIÓN 5-3 CÓDIGO PARA EDITAR Y GUARDAR IMAGEN EN DISPOSITIVO	26
ILUSTRACIÓN 5-4 CÓDIGO CLASE DIMENSIONES	27
ILUSTRACIÓN 5-5 CÓDIGO USANDO LA CLASE DIMENSIONES	27
ILUSTRACIÓN 5-6 CÓDIGO CONTROLADOR SWIPE PARTE 1	28
ILUSTRACIÓN 5-7 CÓDIGO CONTROLADOR SWIPE PARTE 2	29
ILUSTRACIÓN 5-8 CÓDIGO BUSCAR IMAGEN ARAASAC	29
ILUSTRACIÓN 5-9 CÓDIGO ENCARGADO DE HACER EL WEB REQUEST PARA ARAASAC	30
ILUSTRACIÓN 0-1 MAINACTIVITY	- 9 -
ILUSTRACIÓN 0-2 SPLASHACTIVITY	- 9 -
ILUSTRACIÓN 0-3 VISTA AGENDA LISTA	- 10 -
ILUSTRACIÓN 0-4 VISTA AGENDA DÍA	- 10 -
ILUSTRACIÓN 0-5 LISTA CATEGORÍAS	- 10 -
ILUSTRACIÓN 0-6 TAREAS DENTRO DE CATEGORIA	- 10 -
ILUSTRACIÓN 0-7 PASO 1 TAREA	- 10 -
ILUSTRACIÓN 0-8 PASO 2 TAREA	- 10 -
ILUSTRACIÓN 0-9 PASO 3 TAREA	- 10 -
ILUSTRACIÓN 0-10 PASO 1 EDICIÓN TAREA	- 10 -
ILUSTRACIÓN 0-11 PASO 2 EDICIÓN TAREA (SOCIEDAD)	- 10 -
ILUSTRACIÓN 0-12 PASO 2 EDICIÓN TAREA (OBJETOS)	- 10 -
ILUSTRACIÓN 0-13 PASO 2 EDICIÓN TAREA (TIEMPO)	- 10 -
ILUSTRACIÓN 0-14 AJUSTES	- 10 -
ILUSTRACIÓN 0-15 PASO 3 REALIZAR TAREA	- 10 -

INDICE DE TABLAS

TABLA 1 MATRIZ DE TRAZABILIDAD DE REQUISITOS FUNCIONALES.....	12
TABLA 2 CASO DE USO CU-01 - REGISTRAR CONTRASEÑA	- 1 -
TABLA 3 CASO DE USO CU-02 - CAMBIAR CONTRASEÑA	- 2 -
TABLA 4 CASO DE USO CU-03 - BLOQUEAR EDICIÓN	- 2 -
TABLA 5 CASO DE USO CU-04 - CREAR TAREA	- 3 -
TABLA 6 CASO DE USO CU-05 - EDITAR TAREA.....	- 3 -
TABLA 7 CASO DE USO CU-06 – CREAR OBJETO	- 4 -
TABLA 8 CASO DE USO CU-07 – EDITAR OBJETO	- 4 -
TABLA 9 CASO DE USO CU-08 – CREAR PASO	- 5 -
TABLA 10 CASO DE USO CU-09 – EDITAR PASO	- 5 -
TABLA 11 CASO DE USO CU-10 – CREAR SOCIEDAD.....	- 6 -
TABLA 12 CASO DE USO CU-11 – EDITAR SOCIEDAD	- 6 -
TABLA 13 CASO DE USO CU-12 – CREAR CATEGORÍA	- 7 -
TABLA 14 CASO DE USO CU-13 – EDITAR CATEGORÍA	- 7 -
TABLA 15 CASO DE USO CU-14 – REALIZAR TAREA.....	- 8 -

1 Introducción

1.1 *Motivación*

Tras el invento de las aplicaciones móviles, se empezaron a crear distintas aplicaciones que sirvieran como herramientas de apoyo, pero la mayoría de ellas no tiene en cuenta las necesidades de realizar una adaptación para las personas con diversidad funcional intelectual. Hay que tener en cuenta que para las personas con trastornos como el TEA[1] se puede conseguir una inclusión en la sociedad mediante la tecnología y el desarrollo de aplicaciones móviles que los acompañen, y con estas aplicaciones poderse crear una herramienta muy necesaria en su día a día que les ayude a realizar sus tareas y así poder cubrir cualquiera de sus necesidades esenciales, desde lavarse los dientes a hacer la cama.

Debido a las necesidades educativas especiales [2] que este tipo de personas necesitan, es necesaria la creación de una aplicación móvil como herramienta con la que se pueda facilitar la organización de sus tareas y poderles guiar paso a paso para realizarlas. Las personas con este tipo de trastorno en contadas ocasiones necesitan de un apoyo donde se les guíe en su vida cotidiana, por lo que es importante la existencia de este tipo de aplicaciones donde gracias a ellas y a un buen uso de las mismas, se puede conseguir que gocen de una cierta independencia de otras personas que estuvieran controlando las tareas que realizan, ya que en la mayoría de los casos necesitan de un tutor encargado de sus cuidados.

El desarrollo de las aplicaciones móviles ha crecido de manera exponencial los últimos años, pero no en todas las áreas, por lo que es muy positiva la realización de este tipo de proyectos donde se pueda conseguir incluir a las personas con TEA al mundo de la tecnología, la cual les pueda aportar una inclusión social y así poder disminuir las barreras de la comunicación que este tipo de personas suelen sufrir[3]

1.2 *Objetivos*

El principal objetivo es la realización de una aplicación móvil adaptada para personas con diversidad funcional intelectual donde puedan crear, planificar y organizar sus tareas diarias.

Basándose en el objetivo principal, este proyecto se divide en sub-objetivos tales como, la adaptación de la plataforma con pictogramas, el cual es uno de los sub-objetivos más importantes de la plataforma para la cumplimentación de la parte de adaptación del objetivo principal. Con la inclusión de pictogramas en la aplicación conseguimos que las personas con trastornos del espectro autista comprendan mejor las tareas a realizar y así poder simplificar su comprensión.

Otro de los objetivos de esta aplicación es aportar un cierto orden y planificación de la vida diaria del usuario, con la que pueden llegar a conseguir con apartados como, anotar objetos para una tarea, ordenar las tareas por fechas, indicar la duración de una tarea, hacerles ver que comportamientos sociales deben realizar en según qué tareas y darles una pequeña información de los pasos a seguir para poder completar las distintas tareas.

1.3 *Organización de la memoria*

La memoria consta de los siguientes capítulos:

- **Estado del arte:** En este punto se hablará sobre algunas de las aplicaciones existentes, que puedan servir de apoyo para nuestro proyecto.
- **Metodología y análisis de requisitos:** En este apartado se habla sobre el modelo de proceso seguido, se mostrará un diagrama de Gantt sobre el que se ha basado el proyecto y se hará un análisis de los requisitos funcionales y no funcionales.
- **Diseño:** Por último, se muestran las distintas maquetas que realizaron previamente al desarrollo.
- **Desarrollo:** En este punto se indica cual ha sido proceso de codificación de la aplicación y se muestra alguna de los métodos significativos del desarrollo.
- **Integración, pruebas y desarrollo:** Se harán pruebas unitarias que formarán parte de este punto.
- **Conclusiones y trabajo futuro:** En este apartado se incluyen una conclusión sobre el trabajo realizado y los distintos trabajos que se podrían realizar en un futuro en la aplicación.
- **Referencias:** Referencias que han servido como fuente informativa para la realización de este proyecto.
- **Glosario:** Apartado donde se describen brevemente las palabras más importantes del proyecto.
- **Anexos:** Se encuentran en este apartado todas las tablas de los casos de uso, las maquetas y los pantallazos de la aplicación.

2 Estado del arte

2.1 Introducción

Hoy en día existen diversas aplicaciones Android con las que poder planificarte tu día a día o con las que estructurar tus tareas anotando los distintos pasos a realizar. Android cuenta con una plataforma, creada por Google, desde donde poder descargar todas las aplicaciones llamada Google play store[4], anteriormente llamada Android Market, esta plataforma para la descarga de exclusivamente aplicaciones Android, tiene un sistema de puntuación y comentarios de los usuarios que es muy útil para conocer distintas opiniones de otros usuarios que ya la hayan utilizado. Según un estudio realizado por la consultora Appfigures [5], Google play a finales de 2017 contaba en su catálogo con 3,6 millones de aplicaciones móviles. Cifra notablemente superior a su principal competidor AppStore de Apple, que en esa misma fecha contaba con la cifra de 2,1 millones de aplicaciones.

Se ha realizado una búsqueda en la play store de las apps que se puedan utilizar como herramienta auxiliar para la organización y planificaciones de nuestras tareas diarias. A continuación, se muestra un resumen de las aplicaciones que están relacionadas con nuestro objetivo, y de las que nos han podido servir como referente en algunas partes.

2.2 Día a día (organizador)[6]

Aplicación Android que sirve para organizar y planear las tareas, se ha integrado por completo con Google Calendar y Google Task. Un planificador simple con vista mensual, semanal y diaria de texto. Con opciones de utilización de widgets.



Ilustración 2-1:
Día a día

2.3 TimeTune – Optimiza Tu Tiempo[7]

Sirve para organizar el tiempo de forma estructurada añadiendo distintas tareas y programándolas para distintas horas del día. Aplicación simple con iconos simples y estadísticas de los planes realizados cada día con posibilidad de enviar horarios a otras personas.



Ilustración 2-2:
TimeTune

2.4 PictogramAgenda[8]

Agenda visual donde se pueden guardar los distintos planes representados con pictogramas, en ella se realiza una representación visual secuencial de hasta un máximo de 12 pictogramas para ir paso a paso siguiendo las distintas tareas. Tiene colaboración con el portal de imágenes Araasac.



Ilustración 2-3:
PictogramAgenda

2.5 Planificador Semanal[9]

Organizador semanal con widget para Android. Donde se observa una estructuración de las tareas incluyendo distintos campos para la clasificación de las mismas.



Ilustración 2-4:
Planificador Semanal

2.6 AssisT-Task[10]

Herramienta móvil de guiado para la realización de actividades de la vida diaria adaptada a personas con diversidad funcional intelectual. En esta aplicación se puede observar una manera de cómo adaptar la visualización a los pictogramas y como simplificar la aplicación de tal manera que sea rápida y fácil su uso.



Ilustración 2-5:
AssisT-Task

2.7 Conclusiones

Observando las distintas aplicaciones existentes en la actualidad, nos cercioramos de que nuestra aplicación aporta un orden y una estructuración que junto con la adaptación para personas con TEA es inexistente en el mercado, haciendo que esta mezcla de orden y adaptación aporte un salto cualitativo para este tipo de aplicaciones.

La aplicación que se desarrollara en este Trabajo de Fin de Grado juntara algunas de las características y funcionalidades principales de cada una de las aplicaciones expuestas, aportando una adaptación para las personas con diversidad funcional intelectual, además de una estructuración de la aplicación que aporte mayor sencillez y usabilidad.

3 Metodología y análisis de requisitos

3.1 *Modelo de proceso*

Para el diseño de la aplicación he decidido seguir un modelo de refinamiento por pasos o mejora iterativa en cascada, este modelo es una versión con iteraciones del conocido como modelo en cascada o desarrollo en cascada [11] con el he ido refinando y mejorando las distintas fases por las que ha ido pasando el software desarrollado, entre las que se encuentran el análisis de requisitos, el diseño, la codificación y las pruebas e integración. Además, este software pasara por la fase de operación y mantenimiento una vez sea entregado a las personas encargadas de su explotación.

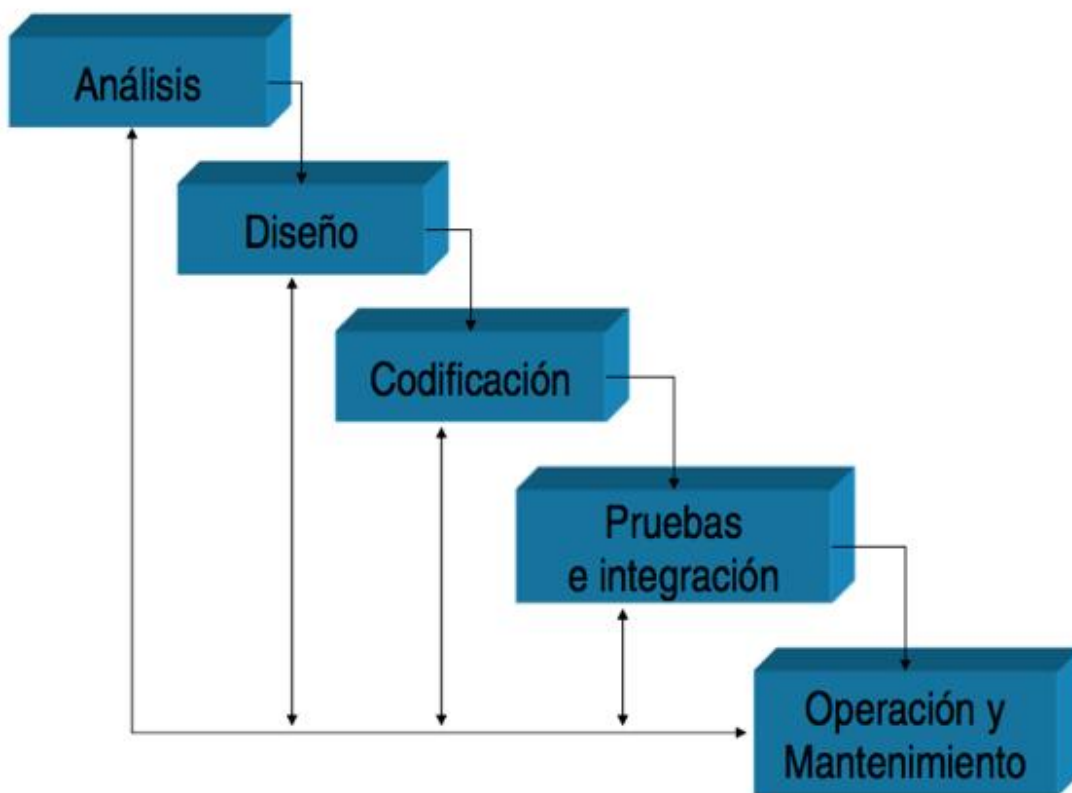


Ilustración 3-1: Modelo de proceso de mejora iterativa en casca

Un modelo de proceso en cascada se realiza de manera lineal donde cada fase comienza cuando ha terminado la anterior, pero en este caso hemos ido realizando iteraciones en cada fase con las que íbamos aportando mayor robustez al proyecto. El modelo en cascada por si solo es incompleto para el desarrollo de una aplicación software, ya que si encontráramos un problema en cualquiera de las fases no podríamos volver a rehacer y mejorar según los problemas encontrados. Pero gracias a hacer este modelo híbrido con las iteraciones conseguimos esa flexibilidad que ayuda a la mejoría constante. A continuación, paso a exponer las distintas fases por las que ha pasado el proyecto, detallando cada una de las iteraciones que se han realizado en cada fase.

Explicación de cada paso del modelo.

- **Análisis de requisitos:** Para esta fase, se realizaron reuniones con la directora y profesora de la escuela ALENTA Asociación Instituto de Psico-Pediatría, en donde se han realizado la toma de requisitos, a partir de los cuales se han ido implantando las siguientes fases. En esta parte surgieron las primeras iteraciones una vez realizada la parte de maquetación en el diseño.
- **Diseño:** En esta fase, inicialmente se realizaron unas primeras maquetas que tras una reunión con el tutor se procedieron a realizar algunas mejoras y surgieron nuevas dudas de mejora para realizar el diseño lo más adaptado a los requisitos iniciales. A partir de las maquetas, se creó una lista de actividades y junto con los requisitos obtenidos en la primera fase se realizó también un diagrama de casos de uso.
- **Codificación/Desarrollo:** Para la fase de codificación o desarrollo, se realizó primeramente un diagrama de clases. Con la lista de actividades obtenida en la fase de diseño, se dividió en una sub-fase por cada actividad.
- **Pruebas e integración:** En cuanto a las pruebas realizadas se han realizado pruebas unitarias, gracias a las cuales se encontraron algunos errores en dos actividades, con las que se realizó una iteración con la fase de codificación de esas actividades y nuevamente se realizó una segunda parte de pruebas unitarias.
- **Mantenimiento:** Esta fase no entra dentro del alcance de este Trabajo de Fin de Grado.

3.2 *Diagrama de Gantt*

Para el correcto cumplimiento de las distintas fases del modelo de proceso se ha realizado un diagrama de Gantt [12], el cual es un método muy eficaz para la gestión de proyectos, que permite reflejar la relación entre las tareas a realizar en el proyecto y como se dividirán en la línea del tiempo. Además, el realizar este tipo de diagramas ayuda para organizar y aclarar más aun las necesidades del proyecto ya que se debe realizar una lista previa de las actividades a desarrollar y las necesidades de las mismas aunque más adelante a la hora de desarrollar siempre aparezcan nuevas partes a incluir y se podrán realizar iteraciones sobre el diseño para incluir estas nuevas actividades.

En el diagrama se puede observar el tipo modelo de proceso comentado, con las distintas iteraciones que se han realizado entre las fases de análisis de requisitos y diseño y las fases de pruebas y codificación. La siguiente ilustración es del diagrama final tras todas las modificaciones que se han ido realizando a la vez que se encontraban errores y se iban realizando las iteraciones. En algunas fases ha habido un pequeño retraso que han sido salvables gracias a que en otras fases ha habido una mejora de los tiempos estipulados, por lo que finalmente no ha habido ninguna repercusión para el cliente final.

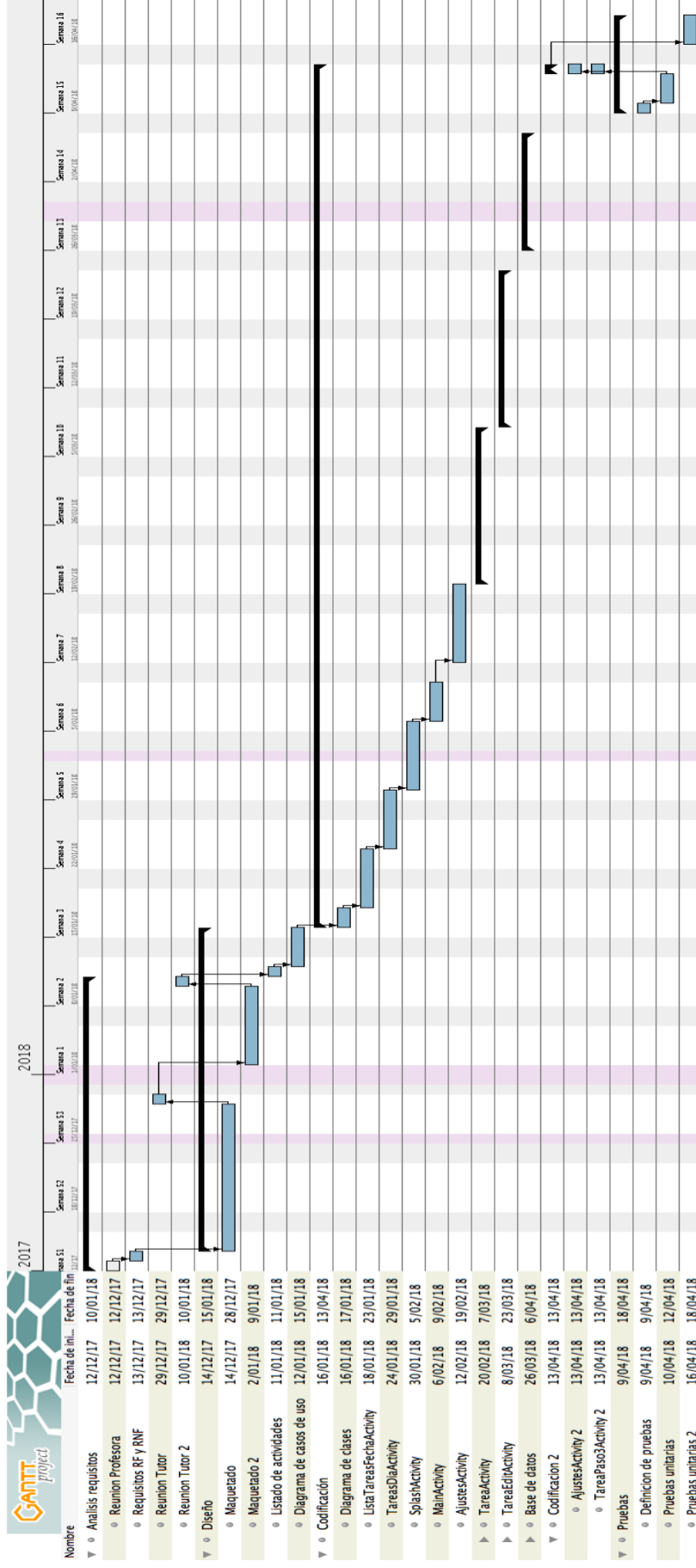


Ilustración 3-2 Diagrama de Gantt

3.3 *Análisis de requisitos funcionales*

- **RF01 - Crear una tarea:** Los usuarios avanzados podrán crear tareas
- **RF02 - Crear una categoría de tareas:** Los usuarios avanzados podrán crear una categoría donde se incluyan tareas de esa categoría.
- **RF03 - Registrar una contraseña:** Los usuarios avanzados deberán registrar una contraseña al inicio de la aplicación.
- **RF04 - Validar con una contraseña al entrar a ajustes:** Los usuarios avanzados deberán validarse con una contraseña para entrar a los ajustes o preferencias.
- **RF05 - Cambiar contraseña:** Los usuarios avanzados deberán tener la opción de cambiar la contraseña.
- **RF06 - Bloquear para que otro usuario no pueda editar:** Los usuarios avanzados podrán bloquear la edición o creación, para que los alumnos no avanzados no puedan editar o crear tareas o categorías.
- **RF07 - Utilizar imágenes del portal de Araasac:** Los usuarios avanzados podrán seleccionar imágenes para las distintas tareas obteniendo pictogramas del portal de Araasac.
- **RF08 - Utilizar imágenes de la cámara del dispositivo:** Los usuarios avanzados podrán seleccionar imágenes para las distintas tareas obteniendo imágenes con su cámara del dispositivo.
- **RF09 - Utilizar imágenes del carrete del dispositivo:** Los usuarios avanzados podrán seleccionar imágenes para las distintas tareas obteniendo imágenes del carrete del dispositivo.
- **RF10 - Añadir fecha a una tarea:** Las tareas tendrán la opción de ponerle una fecha
- **RF11 - Modificar una categoría:** Los usuarios avanzados tendrán que poder modificar una categoría, editando sus atributos internos.
- **RF12 - Modificar una tarea:** Los usuarios avanzados tendrán que poder modificar una categoría, editando sus atributos internos como sus imágenes, título, objetos pasos, comportamientos sociales.
- **RF13 - Añadir objetos a tener para una tarea:** Los usuarios avanzados podrán añadir objetos que fueran necesarios a la hora de realizar una tarea.
- **RF14 - Añadir pasos sociales a tener en cuenta para una tarea:** Los usuarios avanzados podrán añadir costumbre o situaciones sociales que fueran necesarias tener en cuenta a la hora de realizar una tarea.

- **RF15 - Indicar el tiempo estimado de duración de una tarea:** Los usuarios avanzados tendrá que tener la opción de indicar cuál es la duración estimada de una tarea.
- **RF16 - Indicar los pasos a realizar en una tarea:** Los usuarios avanzados podrán añadir los pasos a seguir que fueran necesarios a la hora de realizar una tarea.
- **RF17 - Modificar el orden de los pasos para realizar una tarea:** Los usuarios avanzados deberán tener la opción de modificar el orden de los pasos añadidos que fueran necesarios a la hora de realizar una tarea.
- **RF18 - Seleccionar imagen:** Toda parte de la aplicación deberá estar acompañada de una imagen o pictograma que la represente.
 - **Imagen de Categoría**
 - **Imagen de Paso**
 - **Imagen de Objeto**
 - **Imagen de Tarea**
 - **Imagen de Sociedad**
- **RF19 - Añadir título:** Toda parte de la aplicación deberá estar acompañada de una pequeña descripción o título que la represente.
 - **Título Categoría**
 - **Título Tarea**
 - **Título Paso**
 - **Título Objeto**
 - **Título Sociedad**
- **RF20 - Eliminar:** Toda parte indicada a continuación, deberá poderse eliminar por los usuarios avanzados, siempre y cuando se habilite la edición.
 - **Eliminar tarea**
 - **Eliminar objeto**
 - **Eliminar paso**
 - **Eliminar categoría**
 - **Eliminar sociedad**

- **RF21 - Realizar tarea:** Cualquier usuario podrá realizar una tarea completando las distintas fases de la misma
- **RF22 - Marcar los pasos como realizado:** Cualquier usuario podrá marcar los pasos de una tarea como realizados para completarla.
- **RF23 - Marcar los objetos que se posea:** Cualquier usuario podrá marcar los objetos que tiene en su posesión para la realización de la tarea.

3.4 *Análisis de requisitos no funcionales*

- **RNF01 - Accesibilidad cognitiva:**[13] Se debe cumplir la plena accesibilidad cognitiva.
- **RNF02 - Tiempo aprendizaje:** Tiempo de aprendizaje del sistema menor de 4 horas.
- **RNF03 - Mensajes error:** El sistema debe proporcionar mensajes de error informativos mediante notificaciones o alerts.
- **RNF04 - Pantalla táctil:** El usuario interactuara con el sistema mediante la pantalla táctil.
- **RNF05 - Sistema operativo:** El sistema operativo debe ser Android con un sistema minino de Android 6.0.
- **RNF06 - Seguridad:** Es necesaria una contraseña para poder crear o modificar dentro de la aplicación.
- **RNF07 - Araasac imágenes:** Sera necesaria una clave que provea Araasac para descargar los pictogramas de su plataforma.

4 Diseño

En esta fase, se ha realizado el diseño de la aplicación a partir de los requisitos recogidos en la fase de análisis de requisitos. Se han realizado un diagrama de casos de uso y se ha detallado cada uno de los casos de uso. Además, se presentará una matriz de trazabilidad en la que ver representada la relación entre los requisitos y los casos de uso.

4.1 Casos de uso

4.1.1 Diagrama de casos de uso

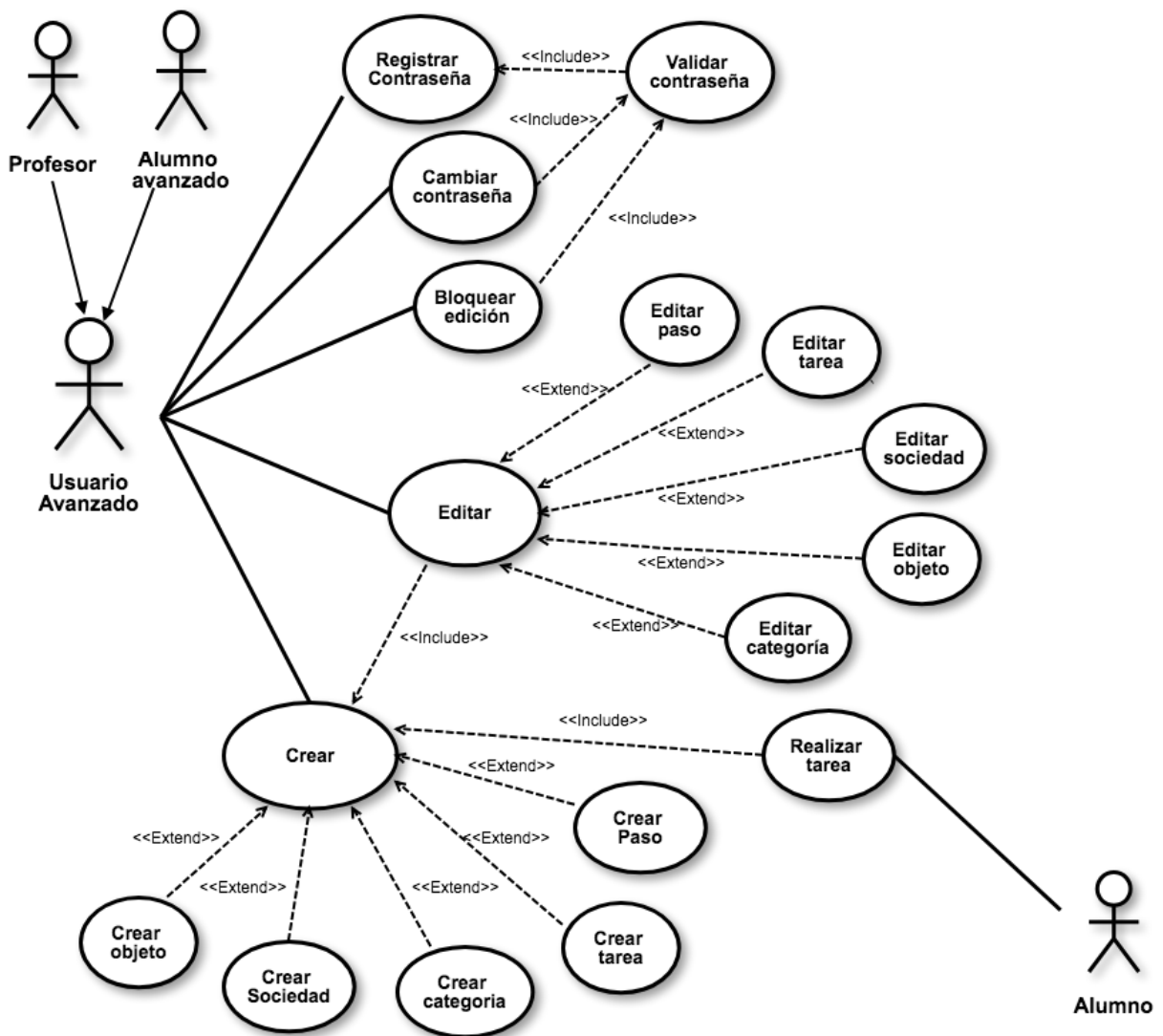


Ilustración 4-1 Diagrama de casos de uso

4.1.2 Descripción de los casos de uso

Se adjuntan las tablas con cada uno de los casos de uso en el Anexo A.

4.1.3 Matriz de trazabilidad con requisitos funcionales

En la siguiente matriz podemos observar una tabla con las correspondencias de los casos de uso con los requisitos funcionales.

REQUISITOS \ CASOS DE USO	CU01	CU02	CU03	CU04	CU05	CU06	CU07	CU08	CU09	CU10	CU11	CU12	CU13	CU14
Crear una tarea – RF01				X										
Crear una categoría de tareas – RF02												X		
Registrar una contraseña – RF03	X													
Validar con una contraseña al entrar a ajustes – RF04		X	X											
Cambiar contraseña – RF05		X												
Bloquear para que otro usuario no pueda editar – RF06			X											
Utilizar imágenes del portal de Araasac – RF07				X	X	X	X	X	X	X	X	X	X	X
Utilizar imágenes de la cámara del dispositivo – RF08				X	X	X	X	X	X	X	X	X	X	X
Utilizar imágenes del carrito del dispositivo – RF09				X	X	X	X	X	X	X	X	X	X	X
Añadir fecha a una tarea – RF10				X										
Modificar una categoría – RF11													X	
Modificar una tarea – RF12				X	X									
Añadir objetos a tener para una tarea – RF13				X	X	X								
Añadir pasos sociales a tener en cuenta para una tarea – RF14				X	X	X				X				
Indicar el tiempo estimado de duración de una tarea – RF15				X	X	X								
Indicar los pasos a realizar en una tarea – RF16				X	X	X		X						
Modificar el orden de los pasos para realizar una tarea – RF17				X	X	X					X			
Seleccionar imagen – RF18				X	X	X	X	X	X	X	X	X	X	X
Añadir título – RF19				X	X	X	X	X	X	X	X	X	X	X
Eliminar – RF20				X	X	X	X	X	X	X	X	X	X	X
Realizar tarea – RF21														X
Marcar los pasos como realizado – RF22														X
Marcar los objetos que se posea – RF23														X

Tabla 1 Matriz de trazabilidad de requisitos funcionales

4.2 Modelo-Vista-Controlador (MVC)

El patrón de arquitectura software que sigue esta aplicación es el del Modelo-Vista-Controlador[14]. Este patrón se fundamenta en la separación de los datos de una aplicación, la interfaz de usuario y la lógica de control en tres partes, el modelo, la vista y el controlador.

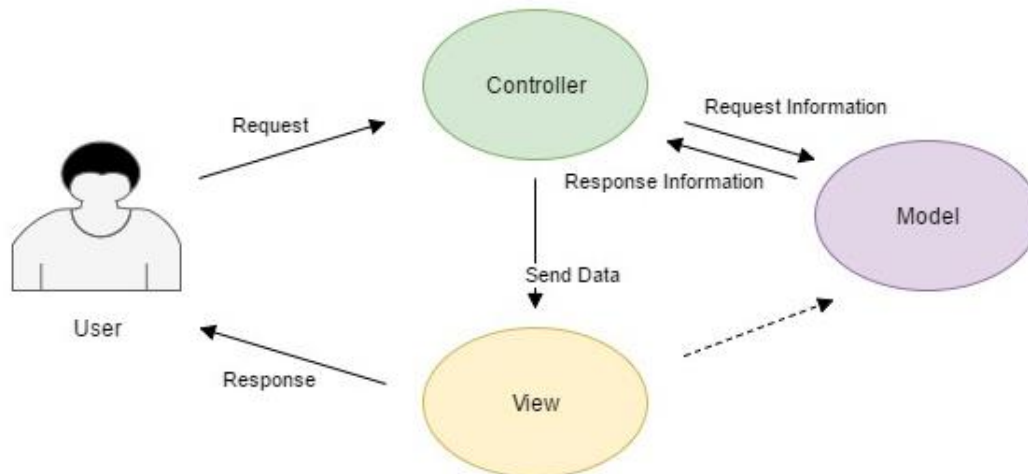


Ilustración 4-2 Modelo-Vista-Controlador

Las responsabilidades de cada parte son:

- **Modelo:** Es la parte encargada de la representación de los datos que serán manejados por el sistema. Contiene mecanismos para el acceso a la información en la base de datos y para ir actualizando sus atributos según le va llegando las actualizaciones a través del controlador. Procesa todos estos datos enviando los datos solicitado por la vista.
- **Vista:** Esta parte se encarga de presentar visualmente los datos aportados por el modelo. Contiene el código que produce la visualización de las interfaces de usuario.
- **Controlador:** Es el encargado de ser el intermediario entre la vista y el modelo, respondiendo a eventos y gestionando el flujo de información que se producen entre ellos.

4.1 Diagrama de clases

A continuación, se muestra el diagrama de clases de la parte correspondiente al modelo de la aplicación, se puede observar que no existe ninguna relación directa entre las distintas clases ya que, están interrelacionadas mediante los identificadores únicos que representan a cada clase. Por ejemplo, para la clase Tarea, se sabe qué pasos, objetos y sociedades pertenecen a ella, gracias a que cada una de las clases tienen el identificador único de la tarea dentro de sus atributos.

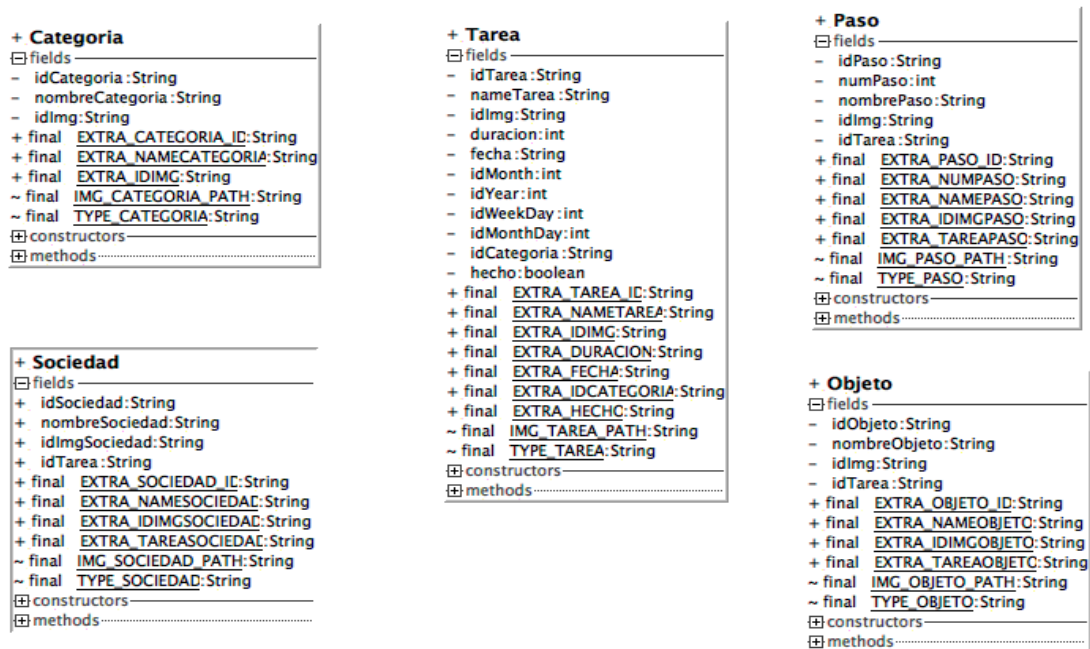


Ilustración 4-3 Diagrama de clases del modelo

- **Tarea:** Esta clase será la clase representara a la información de los datos principales de la aplicación que son las tareas. Es utilizada para crear las distintas tareas de la aplicación, cuenta con atributos como su nombre (*nameTarea*), el identificador único (*idTarea*), el identificador de la imagen que le representa (*idImg*), el indicador de la duración que tardará en realizarse la tarea (*duración*), la fecha para la que estará programada la tarea (*fecha*), el boolean para saber si está realizada o no la tarea (*hecho*) y el identificador de la categoría a la que pertenece la tarea (*idCategoría*).
- **Categoría:** En esta clase se representan los datos de las categorías en las que se pueden dividir las distintas tareas. Entre sus atributos esta un identificador único de categoría (*idCategoría*), un nombre para la categoría (*nombreCategoría*) y un identificador para la imagen que le representa (*idImg*).
- **Sociedad:** Para la representación de los datos la sociedad se ha creado esta clase que contiene los datos que el usuario deberá tener en cuenta sobre cómo comportarse y convivir de una manera socialmente correcta. Entre sus atributos se encuentra el del identificador único de la sociedad (*idSociedad*), el nombre de la sociedad (*nombreSociedad*), el identificador de la imagen que representa a la sociedad (*idImgSociedad*) y el identificador de la tarea a la que pertenece (*idTarea*).
- **Paso:** En esta clase se representan los datos de los pasos a seguir para realizar la tarea de manera correcta. Dentro de sus atributos cuenta con un identificador único de paso (*idPaso*), el indicador del número del paso y en qué posición se debe realizar (*numPaso*), el nombre del paso (*nombrePaso*), el identificador de la imagen que representa al paso (*idImg*) y el identificador de la tarea a la que pertenece el paso (*idTarea*).
- **Objeto:** Para la representación de los datos de los objetos se ha creado la clase Objeto, la cual contendrá los distintos datos de los objetos que sean necesarios para la realización de la tarea a la que pertenezca. Dentro de sus atributos está el del identificador del objeto (*idObjeto*), el nombre del objeto (*nombreObjeto*), el identificador de la imagen que lo representa (*idImg*) y el identificador de la tarea a la que pertenece (*idTarea*).

4.2 Maquetas

Las maquetas son una herramienta esencial para estructurar el proyecto de una manera visual y así facilitar también la propia estructuración de la fase de desarrollo. Ya que a la hora de desarrollar una aplicación móvil necesitas realizar una división de la pantalla mediante diferentes layouts, a partir de una maqueta podemos saber mejor que layouts utilizar y que elementos introducir en cada layout.

“Un layout es un controlador de una o más vistas y que controla su comportamiento y posición”[15]. Durante el diseño y desarrollo de este Trabajo de Fin de Grado se han utilizado los siguientes tipos de Layouts.

- LinearLayout
- TableLayout
- RelativeLayout
- TabLayout

Para la realización de las maquetas que se muestran a continuación se ha utilizado la herramienta online Creatly[16]. Con la que te permite realizar una aproximación de cómo será la vista de las distintas pantallas de la aplicación, aunque no termina de ajustarse al resultado final luego mostrado en los pantallazos de la aplicación ya que no te permite importar el repositorio de material design de Android.

Todas las capturas de las maquetas son la versión final de las mismas, pero todas ellas han pasado por iteraciones entre las fases de diseño y análisis, que tras las reuniones tenidas con la profesora de la escuela y el tutor se han ido mejorando y ajustando más a las necesidades.

4.2.1 Inicio de la aplicación

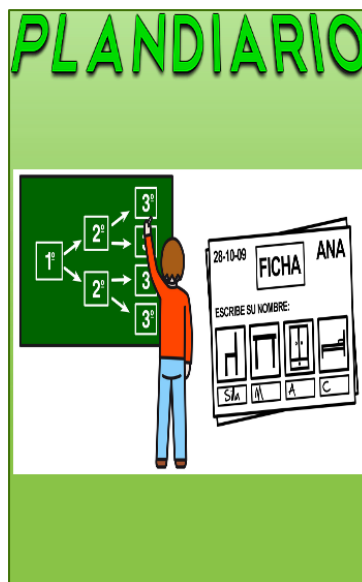


Ilustración 4-4 Maqueta SplashView

Esta maqueta representa la pantalla que se mostrara durante unos segundos para cargar la aplicación y que tras ella se redijera a la pantalla principal de la aplicación.



Ilustración 4-5 Maqueta pantalla principal

Maqueta de la pantalla principal, en donde se muestran las dos opciones de visualización en las que se divide la aplicación. El pictograma de la izquierda (AGENDA) representa la opción de visualizar las tareas que tenga una fecha estipulada. El pictograma de la derecha (CATEGORIAS) representa la opción de ir al listado de categorías de las tareas.

4.2.2 Modo agenda

En este modo existen dos opciones de visualización que tendrán la opción de cambiarse dentro de las preferencias. Estas dos opciones son o en modo lista con todas las tareas de todos los días o en el modo que se muestran las tareas por día.



Ilustración 4-6 Lista tareas modo agenda

Esta maqueta representa cómo será la visualización de las distintas tareas que estén programadas para un día concreto. Donde se observa el pictograma que representa a la tarea, el título y un checkbox que representa si está realizada o no. Existe un Floating Action Button para la opción de crear una tarea nueva.

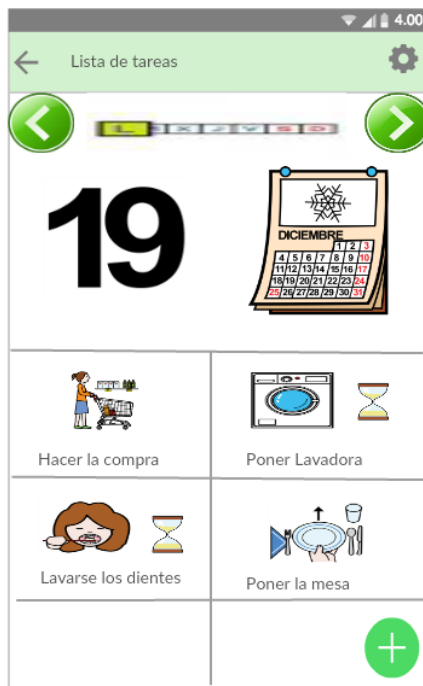


Ilustración 4-7 Visualización de tareas por día en modo agenda

Vista de las tareas que hay programadas para un día concreto, en la que se puede ir modificando el día con las flechas. Se observa una visión en rejilla de las distintas tareas. Existe un Floating Action Button para la opción de crear una tarea nueva.

4.2.3 Modo categoría



Ilustración 4-8 Categorías

Vista de las categorías de una tarea en modo rejilla, con la opción de editar y los ajustes en la toolbar. Existe un Floating Action Button para la opción de crear una categoría nueva. Se observa para cada categoría su pictograma y el título que los representan.



Ilustración 4-9 Lista de tareas dentro de una categoría

En esta vista se puede ver como se muestran las distintas tareas pertenecientes a la categoría que se indica en el título del toolbar. Existe un Floating Action Button para la opción de crear una tarea nueva. Se muestran cada tarea con su pictograma y su título.



Ilustración 4-10 Modo edición de categorías

Esta maqueta representa la vista pensada para la edición de una categoría, donde seleccionando una de las categorías y dándole al botón de editar pasaríamos a editar la categoría.

4.2.4 Visualización de tarea



VAMOS A:

"COMPRAR"



Ilustración 4-11 Paso 1 de tarea

Para el paso 1 se pensó en dividir la pantalla de una manera simple e informativa en la que fácilmente se pudiera ver el título de la tarea y el pictograma que lo representa. También se puede observar que existe la opción de editar desde Toolbar para acceder a la edición de la tarea.



Ilustración 4-12 Paso 2 de tarea

En el paso 2 se divide la pantalla en tres partes, los objetos, los comportamientos con la sociedad y el tiempo. Todas estas partes están representadas con sus pictogramas y un título informativo en la parte izquierda de la pantalla y a la derecha se encuentran los correspondientes ítems que de cada parte. Desde este paso también se puede acceder a la edición de la tarea desde el icono de editar en el Toolbar.

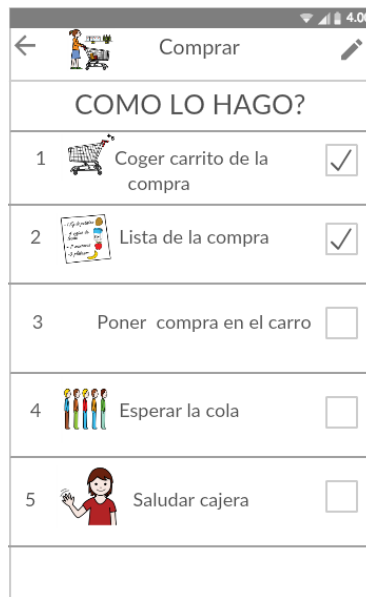


Ilustración 4-13 Paso 3 de tarea

En el paso 3 se representa en la maqueta una lista ordenada de los pasos con un pictograma título y checkbox cada paso. En el Toolbar existe la opción de acceder a la edición de la tarea.

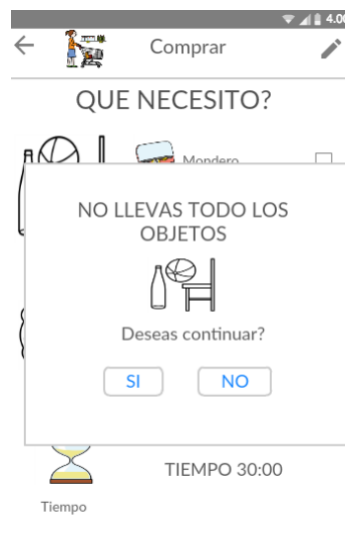


Ilustración 4-14 Notificación de falta de objetos

Con esta maqueta se quiere representar como se muestran las alerts o notificaciones de la aplicación en caso de que se quiera informar al usuario de que le falta algún checkbox por marcar o que ha existido algún error.

4.2.5 Edición y creación de tarea

La edición y la creación de la tarea se pensó en realizar en tres pasos, el primero donde se podían editar las características propias de la tarea, el segundo donde se editarían los objetos, sociedades y el tiempo, y el tercero donde se podrían editar los pasos a realizar en la tarea.



Ilustración 4-15 Paso 1 de editar tarea

En esta ilustración se muestra la maqueta del paso primero de la edición o creación de una tarea donde se puede ver un apartado para la selección de la imagen, otro para el título y otro para seleccionar la categoría a la que pertenecerá la tarea. Además, se muestra un Switch Button donde en caso de estar activo se podrán realizar ajustes temporales. En la parte inferior de la pantalla se encuentra el botón para eliminar la tarea que estemos editando. Para pasar al siguiente paso esta la flecha de la esquina superior derecha.



Ilustración 4-16 Paso 2 de editar tarea

Para el paso segundo de la edición o creación de una tarea, se diseñó una maqueta en la que a través de un TabLayout se pudiera navegar entre las distintas opciones de edición, de izquierda a derecha, de los objetos, sociedades y el tiempo. Para los dos primeros se mostrará una lista y un Floating Button para la creación de objetos o sociedades. Y para el tiempo, un timepicker donde se indique los minutos de duración. En la parte del Toolbar se incluyen el icono para guarda en esta fase o el la flecha para pasar al tercer paso de edición de la tarea.



Ilustración 4-17 Paso 3 de editar tarea

En la maqueta para el paso 3 de la edición o creación de la tarea, se puede observar que los pasos serán mostrados por orden que deben ser realizados. Existe un Floating Action Button con el que se crean nuevos pasos. Para la edición de los pasos existe el icono del lápiz para editar el paso correspondiente. El Toolbar contiene un icono de guardar para finalizar con la edición de la tarea y guardar los cambios realizados.

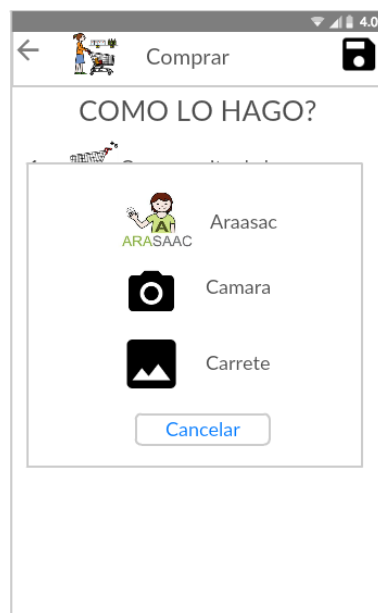


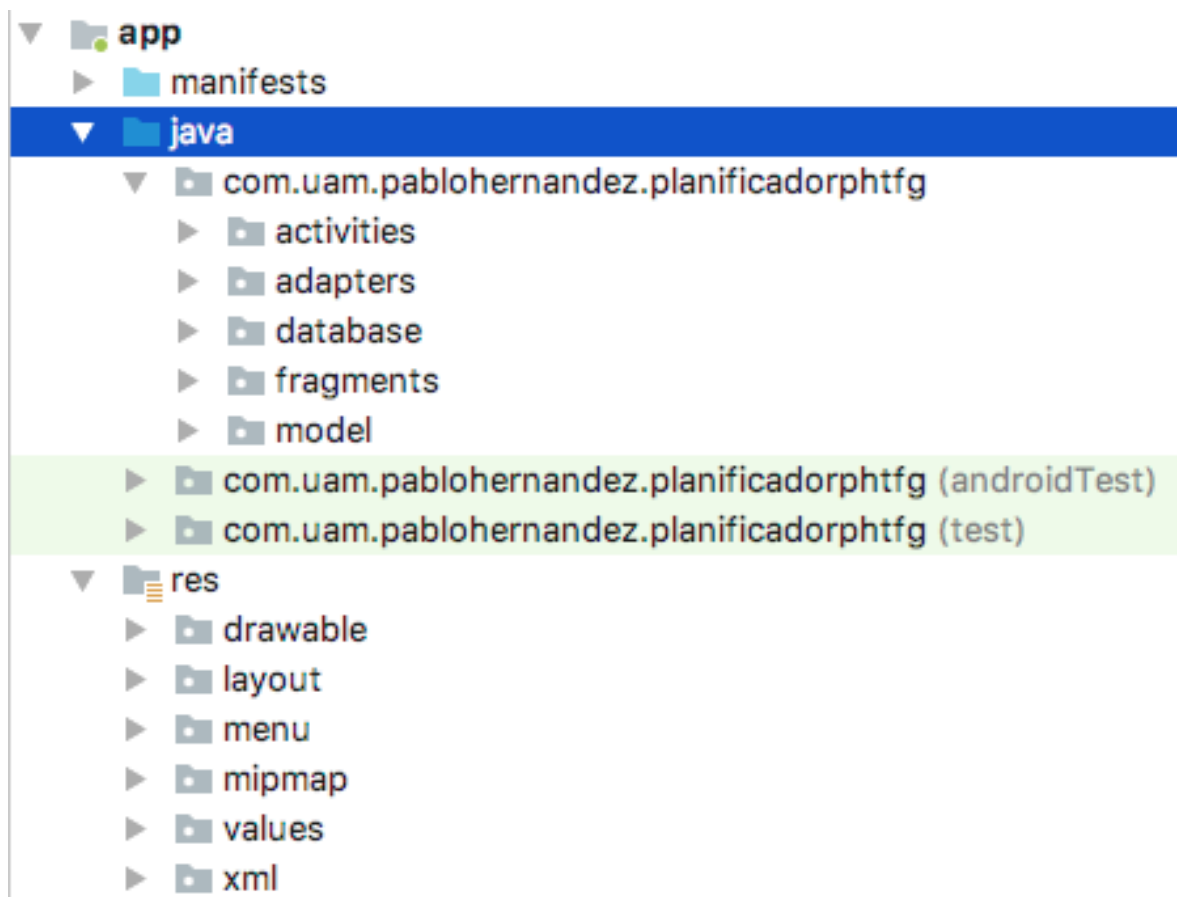
Ilustración 4-18 Selección de imágenes

Para la selección de las imágenes existen tres opciones, desde la plataforma de Araasac, con la cámara del dispositivo y cogiéndola del carrete del dispositivo.

5 Desarrollo

Dentro de este apartado se va a detallar la parte de desarrollo de este Trabajo de Fin de Grado. Hablaré de la base de datos creada para esta aplicación y mostraré un diagrama de la base de datos creada, además, se destacarán algunas partes del código que han sido más complejas o determinantes a la hora de desarrollar la aplicación tanto de la parte del modelo como de la vista. También se hablará de algunos controladores más destacables para el correcto desarrollo de la aplicación.

Como se ha observado en el diagrama de Gantt antes detallado, esta fase ha sido la que mas hora ha llevado terminarla. Toda esta fase, también llamada codificación, ha sido desarrollada en la plataforma para el desarrollo, Android Studio, la cual divide los proyectos en distintas partes, pero en este TFG vamos a entrar a detallar la carpeta `app/src/main/java` donde se encuentran desarrolladas las clases del modelo y la carpeta `app/src/main/res` donde se encuentran los distintos archivos para



- `java`: En esta carpeta está contenido todo el código Java. Dentro se encuentran los distintos package en los que he subdividido todo este código, entre ellos se encuentra el package `activities`, donde están incluidas todas las clases Java encargadas del inflado de las distintas actividades. En el package de `adapters` se encuentran las clases de los adaptadores utilizados en el proyecto, en `fragments` se encuentran las clases de los fragmentos en los que se dividen la pantalla en algunas

actividades. Dentro del package de database, se encuentran todas las clases encargadas de la definición del esquema de la base de datos, de la creación y del acceso y modificación de la misma. Por último, en el package model, se encuentran todas las clases comentadas anteriormente en el apartado de diagrama de clases del modelo.

- res: Esta carpeta contiene todos los archivos necesarios para la obtención de los distintos recursos, como la definición de los colores o cadenas dinámicas, las distintas imágenes que el proyecto necesita o la definición de los distintos layouts que el sistema necesita para mostrar la interfaz al usuario.

5.1 Diagrama de bases de datos

Para la realización del diagrama de bases de datos, se han tenido en cuenta las distintas características necesarias de cada clase del modelo para guardar cada parte importante de la aplicación.

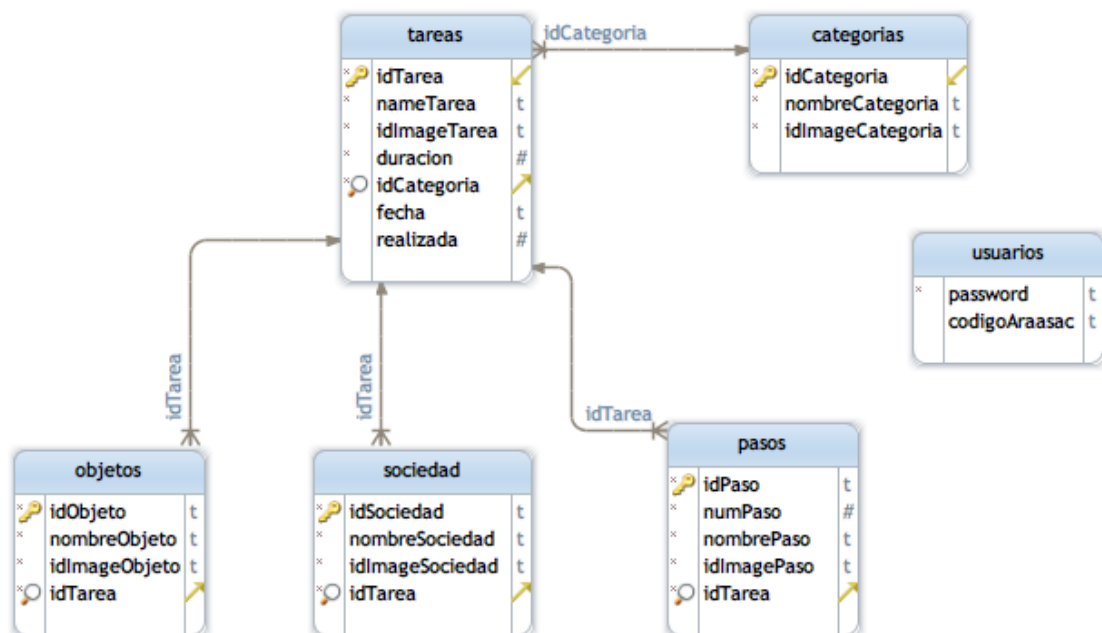


Ilustración 5-1 Diagrama base de datos

En este diagrama, desarrollado con la herramienta DbSchema, se pueden observar cuales son las distintas PrimaryKeys de cada tabla representadas con una llave, además también se representan con una lupa las distintas claves foráneas que tiene cada tabla.

Los más importante a destacar de este diagrama es la relación que tienen cada tabla de objetos, sociedad y pasos con la tabla de tareas, que gracias al identificador de la tarea (idTarea) podemos saber a qué tarea pertenece cada elemento de cada tabla. La tabla tareas también posee un identificador de la Categoría a la que pertenece lo que hace que estén todas interrelacionadas.

La tabla usuarios no tiene una relación con ninguna de las otras tablas, su principal uso es para almacenar la contraseña que registra el usuario al iniciar por primera vez la aplicación y para almacenar el código del portal de Araasac que es necesario para poder descargar pictogramas a nuestro dispositivo.

5.2 Código del modelo

Dentro de la parte de código correspondiente al modelo son destacable 4 funciones que se encargan del guardado, carga, eliminación y edición de las imágenes que se almacenan en los distintos dispositivos.

```
public Bitmap loadImageBitmapFromStorage(Context context)
{
    Bitmap b=null;

    try {ContextWrapper cw = new ContextWrapper(context);
        // path to /data/data/yourapp/app_data/pathDirectory
        File directory = cw.getDir(IMG_TAREA_PATH, Context.MODE_PRIVATE);
        File f=new File(directory, idImg);
        b = BitmapFactory.decodeStream(new FileInputStream(f));
    }
    catch (FileNotFoundException e)
    {
        e.printStackTrace();
    }
    return b;
}

public void deleteImageBitmapFromStorage(Context context) {
    try {
        ContextWrapper cw = new ContextWrapper(context);
        File directory = cw.getDir(IMG_TAREA_PATH, Context.MODE_PRIVATE);
        File f=new File(directory, idImg);
        if(f.exists()) {
            boolean deleted = f.delete();
            if(!deleted)
                throw new Exception();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Ilustración 5-2 Código para cargar y eliminar imagen de dispositivo

En esta ilustración se muestra un pantallazo del código de las funciones que se encargan de guardar o eliminar las imágenes utilizadas dentro de la aplicación. Estas imágenes se almacenan en el directorio /data/data/com.uam.pablohernandez.planificadorphtfg/app_data/tareaImages|categoriaImages|pasoImages|objetoImages|sociedadImages con el directorio final distinto para según a que pertenezca cada imagen.

Con la clase ContextWrapper y su método getDir() podemos acceder al directorio donde tiene almacenadas todos los datos la aplicación.

```

public void changeImageFromTemporalToFinalFromStorage(Context context) {
    try {
        ContextWrapper cw = new ContextWrapper(context);
        File directory = cw.getDir(IMG_TAREA_PATH, Context.MODE_PRIVATE);

        Bitmap bmp = loadImageBitmapFromStorage(context);
        deleteImageBitmapFromStorage(context);
        setIdImgFinal();
        File mypath=new File(directory,idImg);

        FileOutputStream fos = null;
        fos = new FileOutputStream(mypath);
        // Use the compress method on the BitMap object to write image to the OutputStream
        bmp.compress(Bitmap.CompressFormat.PNG, quality: 100, fos);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public String saveToInternalStorage(Bitmap bitmapImage,Context context){
    ContextWrapper cw = new ContextWrapper(context);
    // path to /data/data/yourapp/app_data/pathDirectory
    File directory = cw.getDir(IMG_TAREA_PATH, Context.MODE_PRIVATE);
    // Create imageDir
    String child = idImg;
    File mypath=new File(directory,child);

    FileOutputStream fos = null;
    try {
        fos = new FileOutputStream(mypath);
        // Use the compress method on the BitMap object to write image to the OutputStream
        bitmapImage.compress(Bitmap.CompressFormat.PNG, quality: 50, fos);
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try {
            fos.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    return child;
}
}

```

Ilustración 5-3 Código para editar y guardar imagen en dispositivo

La función `saveToInternalStorage` se encarga de guardar las imágenes en la ruta antes indicada. Y la función `changeImageFromTemporalToFinalStorage` se encarga de modificar el nombre de la imagen guardada como temporal a el nombre de la imagen final. Este proceso es necesario debido a la implementación que se ha desarrollado para la edición de las imágenes de la tarea.

5.3 Código de la vista

El código está desarrollado de tal manera que la interfaz que se muestra al usuario está adaptada al tamaño de cualquier dispositivo, ya que las métricas de las distintas vistas y layouts están definidas con la clase `dimensiones`.

5.3.1 Clase `dimensiones`

Esta clase se encarga de calcular el ancho y el alto para la vista que deseemos crear. Para ello se han implementado dos funciones que se encargan de coger el ancho o el alto de la pantalla del dispositivo y dividirlo por el parámetro que se le pasa según las dimensiones que he estimado que debe ocupar en la pantalla. El uso de esta clase le da mayor dinamismo a la aplicación y así se puede usar en distintos dispositivos sin depender del tamaño del mismo.


```

public class Dimensiones {
    int width=0;
    int height=0;
    Activity activity;

    public Dimensiones(Activity activity){
        this.activity = activity;
        DisplayMetrics displayMetrics = new DisplayMetrics();
        activity.getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);
        width = displayMetrics.widthPixels;
        height = displayMetrics.heightPixels;
        System.out.println("WIDHT: "+width);
        System.out.println("HEIGHT: "+height);
    }

    /**
     * METODO PARA CALCULAR EL ANCHO DE LA PANTALLA QUE NECESITAMOS
     *
     * @param div numero por el que se divide la pantalla
     * @return ancho en pixeles de la pantalla entre div
     */
    public int calculoWidht(int div) { return width/div; }

    /**
     * METODO PARA CALCULAR EL ALTO DE LA PANTALLA QUE NECESITAMOS
     *
     * @param div numero por el que se divide la pantalla
     * @return alto en pixeles de la pantalla entre div
     */
    public int calculoHeight(int div) { return height/div; }
}

```

Ilustración 5-4 Código clase dimensiones

A continuación, se muestra una captura del código donde se utiliza esta clase para indicar el alto y el ancho de una de las imágenes de la vista donde se carga la aplicación.

```

public class SplashActivity extends AppCompatActivity {
    // Set the duration of the splash screen
    private static final long SPLASH_SCREEN_DELAY = 2000;
    Dimensiones dimensiones;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_splash);

        dimensiones = new Dimensiones( activity: this);

        ImageView imgPlani = findViewById(R.id.imgplani);
        RelativeLayout.LayoutParams paramimgPlani = new RelativeLayout.LayoutParams(
            RelativeLayout.LayoutParams.WRAP_CONTENT,
            RelativeLayout.LayoutParams.WRAP_CONTENT);
        paramimgPlani.addRule(RelativeLayout.ALIGN_PARENT_LEFT);

        paramimgPlani.topMargin=dimensiones.calculoHeight( div: 8);
        paramimgPlani.width=dimensiones.calculoWidht( div: 2);
        paramimgPlani.height=dimensiones.calculoHeight( div: 2);

        imgPlani.setLayoutParams(paramimgPlani);
        imgPlani.setAdjustViewBounds(true);
        imgPlani.setScaleType(ImageView.ScaleType.FIT_CENTER);
    }
}

```

Ilustración 5-5 Código usando la clase dimensiones

5.4 Código controlador gesto swipe

Esta clase se crea para controlar el gesto de deslizar el dedo de un lado a otro de la pantalla, también llamado swipe. Se encarga de calcular la posición desde donde hacia donde se desliza el dedo y según en la dirección que este se deslice, se ejecuta un método u otro.

```
public class OnSwipeTouchListener implements OnTouchListener {  
  
    private final GestureDetector gestureDetector;  
  
    public OnSwipeTouchListener (Context ctx){  
        gestureDetector = new GestureDetector(ctx, new GestureListener());  
    }  
  
    @Override  
    public boolean onTouch(View v, MotionEvent event) {  
        return gestureDetector.onTouchEvent(event);  
    }  
  
    public void onSwipeRight() {}  
    public void onSwipeLeft() {}  
    public void onSwipeTop() {}  
    public void onSwipeBottom() {}  
}
```

Ilustración 5-6 Código controlador swipe parte 1

A la hora de crear una clase nueva de OnSwipeTouchListener hay que implementar las funciones onSwipeRight, onSwipeLeft, onSwipeTop y onSwipeBottom que son indicativos de lo que habrá que realizar en el caso de haber deslizado el dedo a cada dirección.

Para la detección del gesto, la clase tiene una clase interna llamada GestureListener que extiende de la clase SimpleOnGestureListener, la cual detecta la posición en el eje X e Y y a partir de esas posiciones y el desarrollo de un algoritmo podemos detectar la dirección en la que se ha realizado el gesto.

```

private final class GestureListener extends SimpleOnGestureListener {
    private static final int SWIPE_THRESHOLD = 100;
    private static final int SWIPE_VELOCITY_THRESHOLD = 100;
    @Override
    public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY) {
        boolean result = false;
        try {
            float diffY = e2.getY() - e1.getY();
            float diffX = e2.getX() - e1.getX();
            if (Math.abs(diffX) > Math.abs(diffY)) {
                if (Math.abs(diffX) > SWIPE_THRESHOLD &&
                    Math.abs(velocityX) > SWIPE_VELOCITY_THRESHOLD) {
                    if (diffX > 0) {
                        onSwipeRight();
                    } else {
                        onSwipeLeft();
                    }
                    result = true;
                }
            }
            else if (Math.abs(diffY) > SWIPE_THRESHOLD &&
                Math.abs(velocityY) > SWIPE_VELOCITY_THRESHOLD) {
                if (diffY > 0) {
                    onSwipeBottom();
                } else {
                    onSwipeTop();
                }
                result = true;
            }
        } catch (Exception exception) {
            exception.printStackTrace();
        }
        return result;
    }

    @Override
    public boolean onDown(MotionEvent e) { return true; }
}

```

Ilustración 5-7 Código controlador swipe parte 2

5.5 Código selección pictogramas Araasac

Para la descarga de pictogramas, se ha realizado una conexión con la API de Araasac con la que podemos acceder a su portal y realizar una búsqueda para la posterior descarga.

```

private void buscar() {
    if(repository.hasAraasacKey()) {
        setURL(language, word, numresul, txtlocate, repository.getAraasacKey());
        new GetImagenes().execute();
    } else {
        AlertDialog.Builder builder = new AlertDialog.Builder( context: AraasacImagenesActivity.this);
        builder.setTitle("Clave Araasac no configurada")
            .setMessage("Debes añadir una clave de Araasac en preferencias")
            .setPositiveButton(android.R.string.ok, (dialog, which) -> {
                dialog.dismiss();
            })
            .setIcon(R.drawable.arasaac)
            .show();
    }
}

```

Ilustración 5-8 Código buscar imagen Araasac

Con la función `setURL()`, formamos la URL siguiendo los parámetros estipulados para realizar peticiones a la API de Araasac, posteriormente ejecutaremos la tarea de búsqueda en segundo plano. En caso de que el usuario no haya indicado en las preferencias la clave para el acceso a los pictogramas, el sistema mostrara una alerta de que debe añadir la clave.

```
private class GetImagenes extends AsyncTask<Void, Void, Void> {

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        progressDialog = new ProgressDialog( context: AraasacImagenesActivity.this);
        progressDialog.setCancelable(true);
        progressDialog.setMessage("Cargando pictos...");
        progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
        progressDialog.setProgress(0);
        progressDialog.show();
    }

    @Override
    protected Void doInBackground(Void... voids) {
        // Creating service handler class instance
        WebRequest webreq = new WebRequest();

        // Making a request to url and getting response
        String jsonStr = webreq.makeWebServiceCall(url, WebRequest.GETRequest);

        Log.d( tag: "Response: ", msg: "> " + jsonStr);

        imgList = ParseJSON(jsonStr);

        return null;
    }

    @Override
    protected void onPostExecute(Void requestresult) {
        super.onPostExecute(requestresult);
        try {
            updateBusqueda();
            progressDialog.dismiss();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Ilustración 5-9 Código encargado de hacer el Web Request para Araasac

Este código muestra como realizamos la llamada al servicio. La función `ParseJSON` se encarga de dividir y extraer las distintas URL de los pictogramas que coinciden con la búsqueda realizada. Y tras el parseo, ejecutamos la función `updateBusqueda()` que actualiza la vista y muestra todos los pictogramas.

5.6 Pantallazos aplicación

Se adjuntan los pantallazos de la aplicación una vez desarrollada toda la codificación en el anexo B.

6 Pruebas y resultados

Una prueba es el proceso de ejecutar un software con el fin de encontrar errores [17]. Se toma como error de software todo aquello que la aplicación no hace y el usuario espera que realice según lo estipulado en los requisitos iniciales. Para encontrar los posibles errores hemos realizado unas pruebas unitarias con un teléfono móvil Asus con sistema operativo Android 6.0 y una Tablet Asus ZD300 con sistema operativo Android 5.0.

La realización de las pruebas que se van a exponer han servido para ir refinando los errores de la aplicación y gracias a las iteraciones realizadas con la parte de codificación o desarrollo, se ha conseguido llegar al objetivo final esperado. Es muy importante la interacción entre las fases de pruebas y desarrollo, ya que ayuda enormemente para la finalización con éxito de ambas fases,

A continuación, se exponen las pruebas realizadas:

- El usuario registra la contraseña.
- El usuario crea una categoría.
- El usuario entra a las preferencias y añade la clave para la plataforma de Araasac.
- El usuario entra a las preferencias y cambia la contraseña.
- El usuario crea una tarea seleccionando la categoría creada, añadiendo un título y seleccionando una imagen con la cámara del dispositivo, añade un objeto, una sociedad, selecciona una duración y añade tres pasos.
- El usuario crea una tarea seleccionando la categoría creada, añadiendo un título y seleccionando una imagen del carrito del dispositivo.
- El usuario crea una tarea seleccionando la categoría creada, añadiendo un título y seleccionando una imagen con la plataforma de Araasac.
- El usuario borra un objeto el editar una tarea.
- El usuario añade una fecha a una tarea y aparece en el modo agenda.
- El usuario borra una tarea.
- El usuario borra una categoría.
- El usuario cambia el nombre de una tarea.
- El usuario cambia el nombre de un objeto.
- El usuario cambia el orden de los pasos de una tarea.
- El sistema marca como realizado cuando el usuario termina una tarea.

Todas las pruebas indicadas, han sido realizadas basándose en los requisitos iniciales estipulados, y en la última iteración han sido pasadas con éxito.

7 Conclusiones y trabajo futuro

7.1 Conclusiones

Tras la realización de las distintas pruebas indicadas, se ha comprobado que la aplicación funciona dando los resultados esperados y pudiéndose realizar las funciones que se espera de la aplicación sin ningún tipo de error.

Con esta aplicación se ha conseguido completar el objetivo inicial, creando una herramienta para la correcta planificación y organización de las personas con diversidad funcional intelectual. Se ha diseñado y desarrollado una interfaz completamente adaptada al usuario final, con la que puedan verse incluidos en la sociedad y superar las posibles barreras de comunicación que les surjan.

Para cubrir el objetivo principal se ha estructurado la aplicación de una manera correcta basándose en Tareas que representan cada una de las tareas diarias que son necesarias para cubrir las necesidades de los usuarios finales. Gracias a este proyecto he conocido cuales son los mejores métodos de tratar a las personas con trastornos como el TEA.

Tras el desarrollo de todas las fases por las que ha pasado este Trabajo de Fin de Grado, se puede concluir que la aplicación esta lista para ser incluida en la plataforma de distribución digital Google Play Store.

7.2 Trabajo futuro

Aunque los objetivos comentados al inicio del proyecto se han cumplido con el desarrollo de esta aplicación, en toda aplicación siempre existe alguna mejora del software, y más aún debido a que es una aplicación creada de cero, siempre se pueden realizar mejoras que van apareciendo al termino del desarrollo de la misma.

Algunas de estas mejoras son:

- **Backup:** Seria una mejora destacable, la realización de un backup de toda la información que almacena la aplicación y así en caso de pérdida o cambio de dispositivo poder mantener la aplicación tal cual la queremos[18].
- **Almacenamiento cloud:** En caso de necesitar una ampliación para almacenar las distintas tareas, el almacenamiento cloud es una método novedoso y sencillo para ampliar el almacenamiento del dispositivo[19].
- **Desbloquear ajustes con huella dactilar:** Una opción para aportar seguridad a las configuraciones de las preferencias es el uso de la huella dactilar cada vez más incluida en aplicaciones[20].
- **Cronometro para tareas:** Incluir un cronometro que tenga en cuenta el momento en que empieza a realizar la tarea y te avise cuando deberías de haberla terminado.
- **Notificaciones de recordatorio de tareas:** Recordar al usuario cuando tiene programada una tarea que debe realizar.

Referencias

- [1] PROYECTO AUTISME LA GARRIGA. (s.f.). *www.autismo.com*. Obtenido de <http://www.autismo.com.es/autismo/que-es-el-autismo.html>
- [2] Universidad Internacional de Valencia. (1 de 10 de 2014). *Universidad Internacional de Valencia*. Obtenido de Las necesidades educativas de los niños en tratamiento por autismo: <https://www.universidadviu.es/las-necesidades-educativas-de-los-ninos-en-tratamiento-por-autismo/>
- [3] NIDCD. (s.f.). Obtenido de Trastornos del espectro autista: Problemas de comunicación en los niños: <https://www.nidcd.nih.gov/es/espanol/problemas-de-comunicacion-en-los-ninos-con-trastornos-del-espectro-autista>
- [4] Google. (s.f.). *Google Play Store*. Obtenido de <https://play.google.com/store/apps?hl=es>
- [5] Aplicantes. (23 de Abril de 2018). Aplicantes. Obtenido de Google Play y la App Store compartieron 450.000 apps en 2017: <http://aplicantes.com/apps-publicadas-ios-android-2017/>
- [6] Leonidovich, A. (2 de Julio de 2016). *Google Play Store*. Obtenido de Día a día (organizador): <https://play.google.com/store/apps/details?id=ru.infteh.organizer.full&hl=es>
- [7] TimeTune. (s.f.). *Google Play Store*. Obtenido de TimeTune - Optimiza Tu Tiempo: <https://play.google.com/store/apps/details?id=com.gmail.jmartindev.timetune&hl=es>
- [8] kilobytelab. (s.f.). *Google Play Store*. Obtenido de Agenda de Pictogramas Smeak: <https://play.google.com/store/apps/details?id=smeak.kilobytelab.agenda&hl=es>
- [9] AimNovate. (s.f.). *Google Play Store*. Obtenido de Planificador Semanal: <https://play.google.com/store/apps/details?id=com.aimnovate.weekyplanner&hl=es>
- [10] Gomez, J., Torrado, J. C., & Montoro, G. (s.f.). Obtenido de Using Smartphones to Assist People with Down Syndrome in Their Labour Training and Integration: A Case Study: <https://www.hindawi.com/journals/wcmc/2017/5062371/>
- [11] Wikipedia. (s.f.). Obtenido de Desarrollo en cascada: https://es.wikipedia.org/wiki/Desarrollo_en_cascada#Variantes
- [12] OBS. (s.f.). *bussines School*. Obtenido de ¿Qué es un diagrama de Gantt y para qué sirve?: <https://www.obs-edu.com/es/blog-project-management/diagramas-de-gantt/que-es-un-diagrama-de-gantt-y-para-que-sirve>
- [13] Plena Inclusion. (06 de Septiembre de 2017). Obtenido de 11 servicios para incorporar la accesibilidad cognitiva en tu institución o empresa: <http://www.plenainclusion.org/informate/actualidad/noticias/2017/11-servicios-para-incorporar-la-accesibilidad-cognitiva-en-tu>
- [14] Hernández, U. (s.f.). *Códigofacilito*. Obtenido de MVC (Model, View, Controller) Explicado: <https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>
- [15] Universidad Politecnica de Valencia. (s.f.). Obtenido de Layouts: <http://www.androidcurso.com/index.php/tutoriales-android/32-unidad-2-diseno-de-la-interfaz-de-usuario-vistas-y-layouts/114-layouts>
- [16] Creatly. (s.f.). Obtenido de Creatly: <https://creately.com/app/>
- [17] it-Mentor. (s.f.). Obtenido de Capacitación y guía para el desarrollo de software: <http://materias.fi.uba.ar/7548/PruebasSoftware.pdf>
- [18] Ortega, L. (Septiembre de 2017). *ANDROIDPIT*. Obtenido de Como hace backup en Android: <https://www.androidpit.es/como-hacer-backup-android>
- [19] Amazon. (s.f.). *AWS Amazon*. Obtenido de Capa gratuita de AWS: https://aws.amazon.com/es/free/?sc_channel=PS&sc_campaign=acquisition_ES&sc_publisher=google&sc_medium=cloud_computing_b&sc_content=aws_cloud_p&sc_detail=almacenamiento%20en%20la%20nube&sc_category=cloud_computing&sc_segment=74598913401&sc_matchtype=p&sc
- [20] Android Developers. (s.f.). *Android Developers*. Obtenido de Android 6.0 API : <https://developer.android.com/about/versions/marshmallow/android-6.0?hl=ES>
- [21] Terrazas, M., Sanchez, S., & Becerra, M. (s.f.). *Dialnet*. Obtenido de Las TIC como herramienta de apoyo para personas con Trastorno del Espectro Autista (TEA): <https://dialnet.unirioja.es/servlet/articulo?codigo=5600282>
- [22] El blog de cerqana. (23 de Enero de 2017). Obtenido de COMUNICACIÓN, TEA Y PICTOGRAMAS: <https://blog.cerqana.com/comunicacion-tea-y-pictograma/>

Glosario

API	Application Programming Interface
TEA	Trastorno del Espectro Autista
FAB	Floating Action Button
URL	Uniform Resource Locator
JSON	JavaScript Object Notation
ARAASAC	Portal Aragonés de Comunicación Aumentativa y Alternativa

Anexos

A Casos de uso

ID	CU-01
NOMBRE	Registrar contraseña
Actor Principal	Usuario avanzado
Actores Secundarios	Ninguno
Descripción	Registro de la contraseña para el control del acceso a las preferencias de la aplicación
Precondición	Primera vez que se inicia la aplicación
Garantía de éxito	El usuario recibirá una notificación de nueva contraseña creada.
Escenario principal	1.Se inicia la aplicación 2.El actor principal introduce contraseña a crear 3.El sistema registra la contraseña
Extensiones	2.El actor principal introduce mal la contraseña 2a. El sistema notifica que ha introducido mal la contraseña 2b. El sistema vuelve a pedir la contraseña 2c. El caso de uso continua con normalidad

Tabla 2 Caso de uso CU-01 - Registrar contraseña

ID	CU-02
NOMBRE	Cambiar contraseña
Actor Principal	Usuario avanzado
Actores Secundarios	Ninguno
Descripción	Cambiar la contraseña del control del acceso a las preferencias de la aplicación
Precondición	Haberse creado una contraseña y validarse con la contraseña actual al entrar a las preferencias
Garantía de éxito	El usuario recibirá una notificación de contraseña cambiada y no se podrá acceder con la antigua contraseña
Escenario principal	1. Usuario accede a las preferencias. 2. Usuario realiza la validación de contraseña. 3. Usuario pulsa en "Cambiar contraseña" dentro de las preferencias. 4. Usuario introduce nueva contraseña.

Extensiones	2.El actor principal introduce mal la contraseña 2a. El sistema notifica que ha introducido mal la contraseña 2b. El sistema vuelve a pedir la contraseña 2c. El caso de uso continua con normalidad
-------------	---

Tabla 3 Caso de uso CU-02 - Cambiar contraseña

ID	CU-03
NOMBRE	Bloquear edición
Actor Principal	Usuario avanzado
Actores Secundarios	Ninguno
Descripción	Bloquear las opciones de edición y creación tanto de tareas como de categorías.
Precondición	Validarse con la contraseña al entrar a las preferencias
Garantía de éxito	Checkbox con marca de checkeado y no se muestran las opciones de edición y creación en ninguna pantalla.
Escenario principal	1. Usuario accede a las preferencias. 2. Usuario realiza la validación de contraseña. 3. Usuario pulsa en el checkbox de "Bloquear edición"
Extensiones	2.El actor principal introduce mal la contraseña 2a. El sistema notifica que ha introducido mal la contraseña 2b. El sistema vuelve a pedir la contraseña 2c. El caso de uso continua con normalidad

Tabla 4 Caso de uso CU-03 - Bloquear edición

ID	CU-04
NOMBRE	Crear tarea
Actor Principal	Usuario avanzado
Actores Secundarios	Ninguno
Descripción	Creación de una tarea
Precondición	Tener habilitada la opción de edición
Garantía de éxito	Se muestra por pantalla la representación de la tarea con su título y su imagen.
Escenario principal	1. El usuario selecciona el icono con el símbolo de añadir (+) 2. El usuario añade un titulo 3. El usuario selecciona una imagen 4. El usuario selecciona una categoría

	5. El usuario selecciona una fecha 6. El usuario crea un objeto 7. El usuario crea una sociedad 8. El usuario selecciona un tiempo de duración 9. El usuario crea un paso 10. El usuario pulsa en el icono de guardar
Extensiones	X. El usuario comete un error Xa. Se informa con una notificación del error Xb. Se continua con el caso de uso con normalidad

Tabla 5 Caso de uso CU-04 - Crear tarea

ID	CU-05
NOM+E18BRE	Editar tarea
Actor Principal	Usuario avanzado
Actores Secundarios	Ninguno
Descripción	Edición de una tarea
Precondición	Tener habilitada la opción de edición, haber creado una tarea y tras ello acceder a la visualización de la tarea
Garantía de éxito	Se muestran los cambios en la visualización de la tarea
Escenario principal	1. El usuario selecciona el icono con el símbolo de editar en el toolbar 2. El usuario edita el título de la tarea 3. El usuario pulsa en el icono de guardar
Extensiones	2. El usuario pulsa el icono de eliminar la tarea. 2a. El sistema retorna al menú principal 2b. Finaliza el caso de uso 2. El usuario edita los pasos 2. El usuario edita los objetos 2. El usuario edita los pasos de la sociedad 2. El usuario edita el tiempo

Tabla 6 Caso de uso CU-05 - Editar tarea

ID	CU-06
NOMBRE	Crear objeto
Actor Principal	Usuario avanzado
Actores Secundarios	Ninguno
Descripción	Creación de un objeto que llevar para una tarea
Precondición	Haber iniciado la creación o edición de una tarea y accedido al segundo paso de la creación o edición.
Garantía de éxito	Se muestran los cambios en la visualización de la tarea
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona el Floating Action Button con el símbolo mas (+) correspondiente a la pestaña de objetos 2. El usuario añade el titulo 3. El usuario selecciona una imagen 4. El usuario guarda el objeto con el símbolo de guardar del toolbar
Extensiones	

Tabla 7 Caso de uso CU-06 – Crear objeto

ID	CU-07
NOMBRE	Editar objeto
Actor Principal	Usuario avanzado
Actores Secundarios	Ninguno
Descripción	Edición de las características de un objeto perteneciente a una tarea
Precondición	Haber iniciado la edición de una tarea y accede al segundo paso de la edición
Garantía de éxito	Se muestran los cambios en la visualización de la tarea
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona el icono con el símbolo de editar de la lista de objetos 2. El usuario edita el titulo o la imagen 3. El usuario pulsa en el icono de guardar
Extensiones	<ol style="list-style-type: none"> 1. El usuario pulsa el icono de eliminar en la descripción del objeto dentro de la lista de objetos <ol style="list-style-type: none"> 1a. El sistema actualiza la lista de objetos 2. El usuario pulsa el icono de eliminar en la edición del objeto <ol style="list-style-type: none"> 2a. El sistema actualiza la lista de objetos

Tabla 8 Caso de uso CU-07 – Editar objeto

ID	CU-08
NOMBRE	Crear paso
Actor Principal	Usuario avanzado
Actores Secundarios	Ninguno
Descripción	Creación de un paso a seguir para completar la tarea
Precondición	Haber iniciado la creación o edición de una tarea y accedido al tercer paso de la creación o edición.
Garantía de éxito	Se muestran los cambios en la visualización de la tarea
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona el Floating Action Button con el símbolo mas (+) 2. El usuario añade el titulo 3. El usuario selecciona una imagen 4. El usuario guarda el paso con el símbolo de guardar del toolbar 5. El sistema actualiza la lista de pasos
Extensiones	

Tabla 9 Caso de uso CU-08 – Crear paso

ID	CU-09
NOMBRE	Editar paso
Actor Principal	Usuario avanzado
Actores Secundarios	Ninguno
Descripción	
Precondición	Haber iniciado la creación o edición de una tarea
Garantía de éxito	Se muestran los cambios en la visualización de la tarea
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona el icono con el símbolo de editar de la lista de pasos 2. El usuario edita el titulo o la imagen 3. El usuario pulsa en el icono de guardar
Extensiones	<ol style="list-style-type: none"> 1. El usuario pulsa el icono de eliminar en la descripción del paso <ol style="list-style-type: none"> 1a. El sistema actualiza la lista de pasos 2. El usuario pulsa el icono de eliminar en la edición del paso <ol style="list-style-type: none"> 2a. El sistema actualiza la lista de pasos

Tabla 10 Caso de uso CU-09 – Editar paso

ID	CU-10
NOMBRE	Crear sociedad
Actor Principal	Usuario avanzado
Actores Secundarios	Ninguno
Descripción	Crear sociedad
Precondición	Haber iniciado la creación o edición de una tarea y accedido al segundo paso de la creación o edición.
Garantía de éxito	Se muestran los cambios en la visualización de la tarea
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona el Floating Action Button con el símbolo mas (+) correspondiente a la pestaña de sociedad 2. El usuario añade el titulo 3. El usuario selecciona una imagen 4. El usuario guarda la sociedad con el símbolo de guardar del toolbar 5. El sistema actualiza la lista de sociedades

Tabla 11 Caso de uso CU-10 – Crear sociedad

ID	CU-11
NOMBRE	Editar sociedad
Actor Principal	Usuario avanzado
Actores Secundarios	Ninguno
Descripción	Edición de las características de un paso de la sociedad perteneciente a una tarea
Precondición	Haber iniciado la edición de una tarea y accede al segundo paso de la edición
Garantía de éxito	Se muestran los cambios en la visualización de la tarea
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona el icono con el símbolo de editar de la lista de sociedades 2. El usuario edita el titulo o la imagen 3. El usuario pulsa en el icono de guardar
Extensiones	<ol style="list-style-type: none"> 1. El usuario pulsa el icono de eliminar en la descripción de la sociedad dentro de la lista de sociedades <ol style="list-style-type: none"> 1a. El sistema actualiza la lista de sociedades 2. El usuario pulsa el icono de eliminar en la edición de la sociedad <ol style="list-style-type: none"> 2a. El sistema actualiza la lista de sociedades

Tabla 12 Caso de uso CU-11 – Editar sociedad

ID	CU-12
NOMBRE	Crear categoría
Actor Principal	Usuario avanzado
Actores Secundarios	Ninguno
Descripción	Creación de una categoría de tarea para la organización de las tareas
Precondición	Acceder a la opción de categorías
Garantía de éxito	Se muestran los cambios en la visualización de la tarea
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona el Floating Action Button con el símbolo mas (+) 2. El usuario añade el titulo 3. El usuario selecciona una imagen 4. El usuario guarda el paso con el símbolo de guardar del toolbar 5. El sistema actualiza la lista de categorías
Extensiones	

Tabla 13 Caso de uso CU-12 – Crear categoría

ID	CU-13
NOMBRE	Editar categoría
Actor Principal	Usuario avanzado
Actores Secundarios	Ninguno
Descripción	Editar categoría
Precondición	Acceder a la opción de categorías
Garantía de éxito	Se muestran los cambios en la visualización de la tarea
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona el icono con el símbolo de editar en el toolbar 2. El usuario pulsa en la categoría a editar 3. El usuario selecciona el Floating Action Button con el símbolo de editar 4. El usuario edita el titulo 5. El usuario pulsa en el icono de guardar
Extensiones	<ol style="list-style-type: none"> 4. El usuario edita la imagen seleccionada 4. El usuario pulsa el icono de eliminar <ol style="list-style-type: none"> 4a. El sistema retorna al menú de categorías 4b. Finaliza el caso de uso

Tabla 14 Caso de uso CU-13 – Editar categoría

ID	CU-14
NOMBRE	Realizar tarea
Actor Principal	Alumno
Actores Secundarios	Usuario avanzado
Descripción	Realizar una tarea planificada
Precondición	Haber sido creada la tarea por el usuario avanzado y accedido a la visualización de la tarea
Garantía de éxito	Checkbox de tarea realizada marcado
Escenario principal	<ol style="list-style-type: none"> 1. El usuario accede al según paso de la tarea 2. El usuario selecciona los checkbox de los objetos que posee 3. El usuario accede al tercer paso de la tarea. 4. El usuario selección los checkbox de los pasos realizados por orden. 5. El usuario selección el icono de la flecha. 6. El sistema retorna a la lista de tareas seleccionado inicialmente 7. El sistema marca la tarea como realizada
Extensiones	<ol style="list-style-type: none"> 3. El usuario no selecciona todos los checkbox de los objetos <ol style="list-style-type: none"> 3a. El sistema informa de que no lleva todos los objetos 3b. El caso de uso continua en el paso 2. 5. El usuario no selecciona todos los checkbox de los pasos <ol style="list-style-type: none"> 3a. El sistema informa de que no lleva todos los pasos 3b. El caso de uso continua en el paso 4.

Tabla 15 Caso de uso CU-14 – Realizar tarea

B Pantallazos aplicación

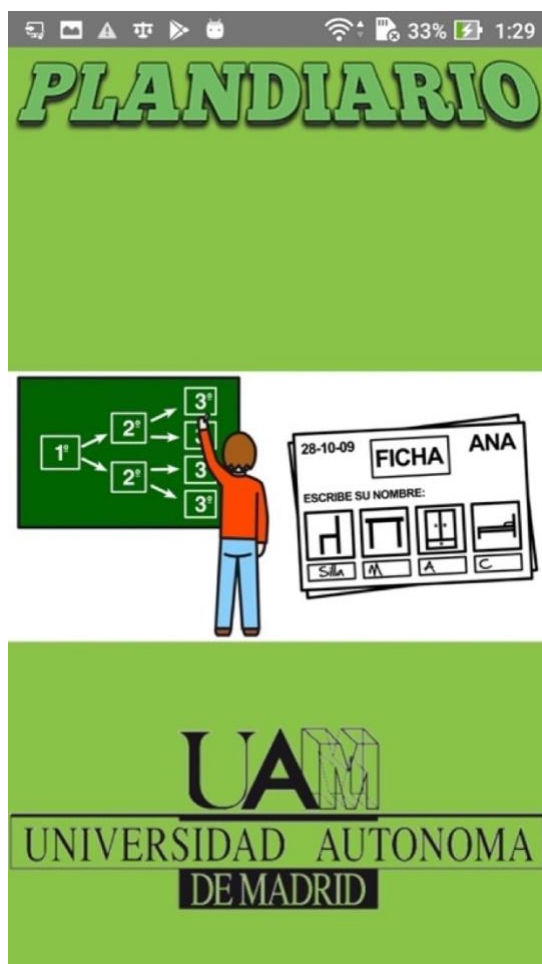


Ilustración 0-2 SplashActivity



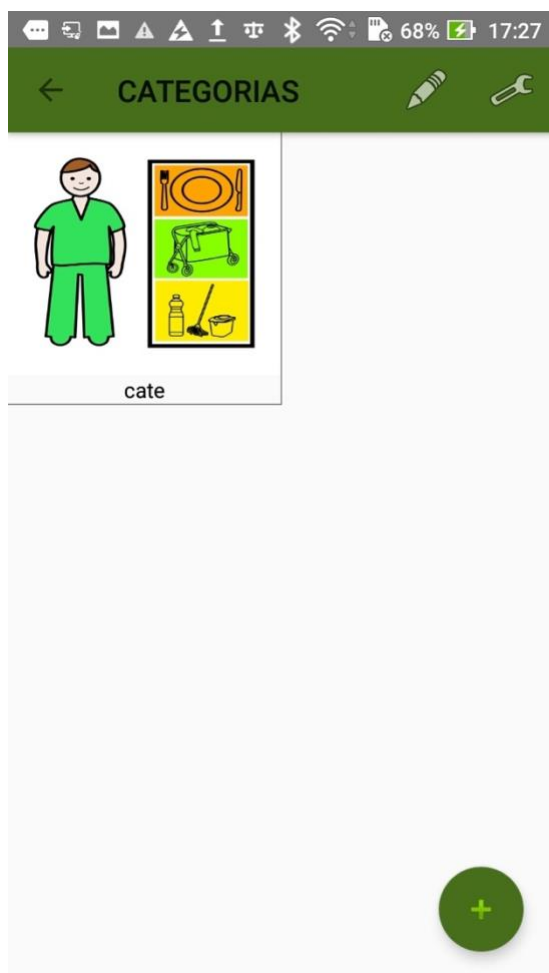
Ilustración 0-1 MainActivity



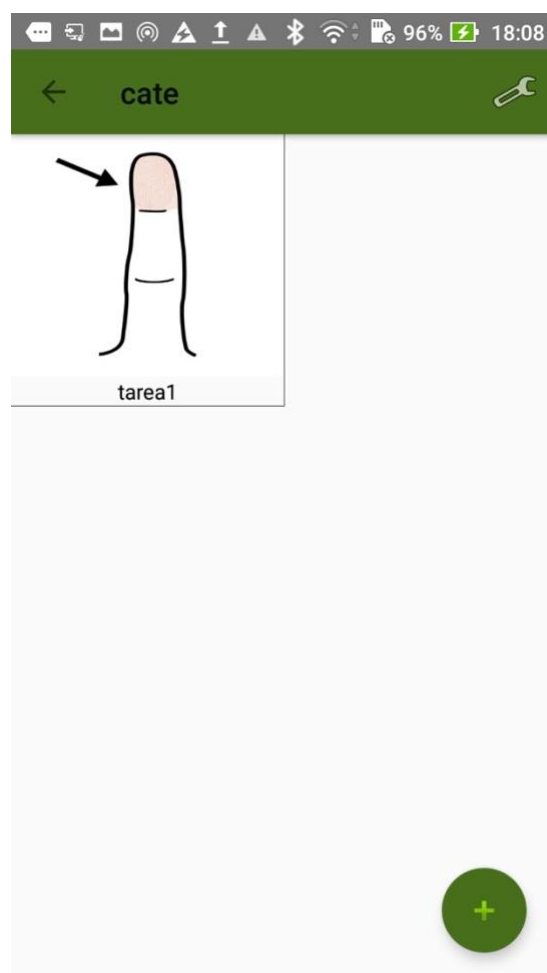
**Ilustración 0-3 Vista
Agenda Lista**



**Ilustración 0-4 Vista
Agenda Día**



**Ilustración 0-6 Lista
Categorías**



**Ilustración 0-5 Tareas
dentro de categoria**

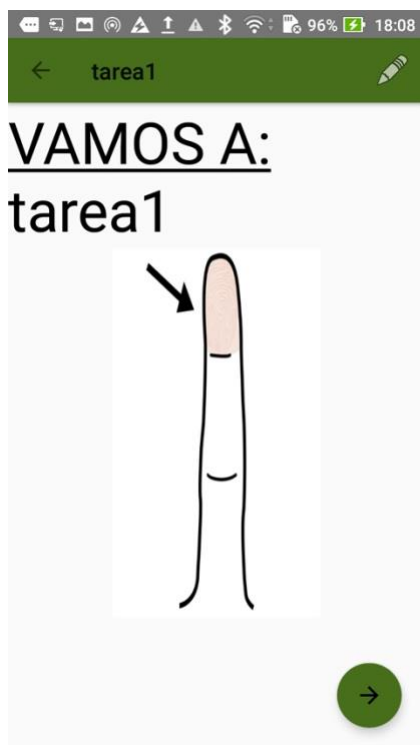


Ilustración 0-7 Paso 1 Tarea

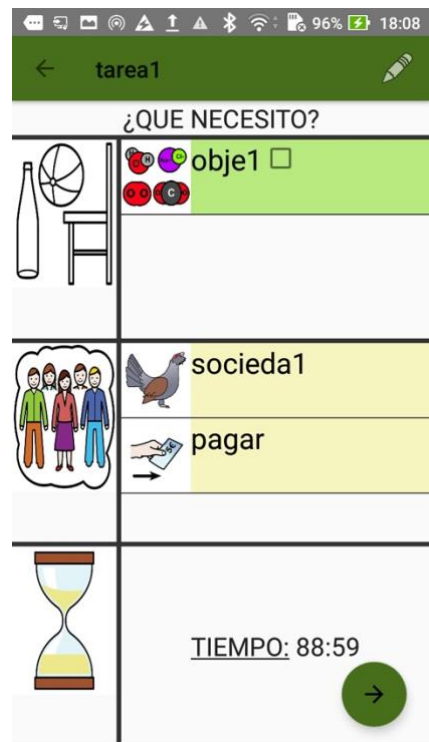


Ilustración 0-8 Paso 2 Tarea

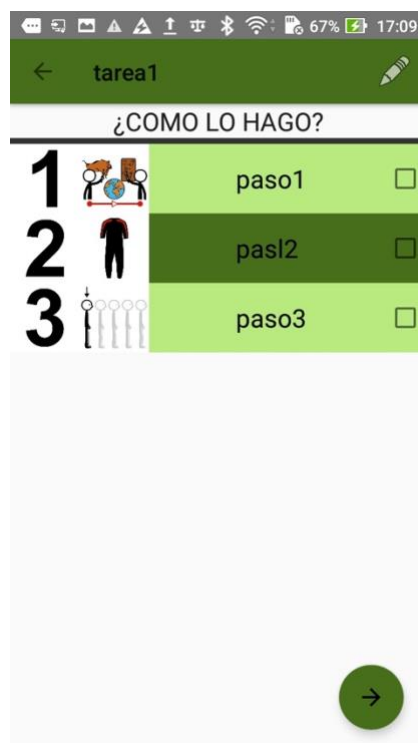
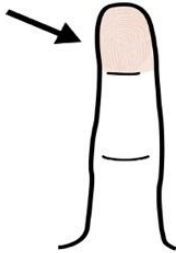


Ilustración 0-9 Paso 3 Tarea



← plandiarioright>

VAMOS A tarea1

Selecciona la imagen:



Selecciona la Categoría:



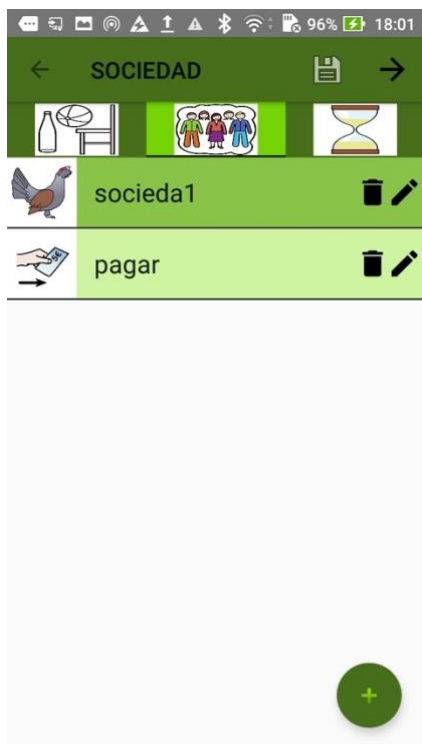
Ajuste temporal ☒

23/5/2018

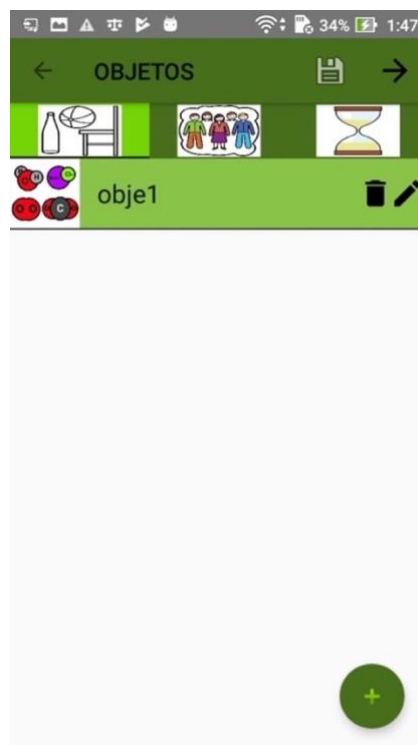
HECHO: ☐



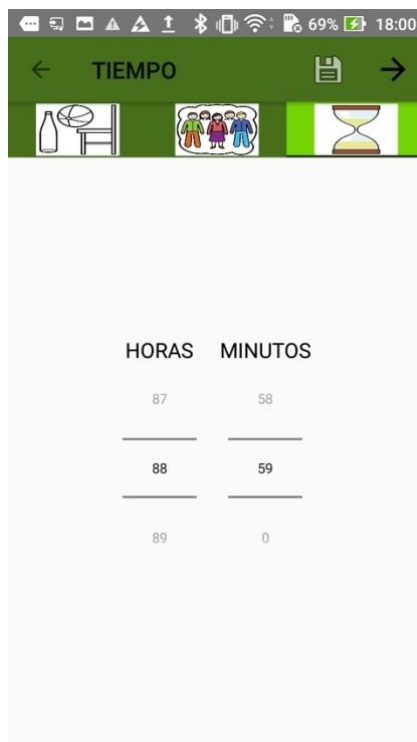
Ilustración 0-10 Paso 1
Edición Tarea



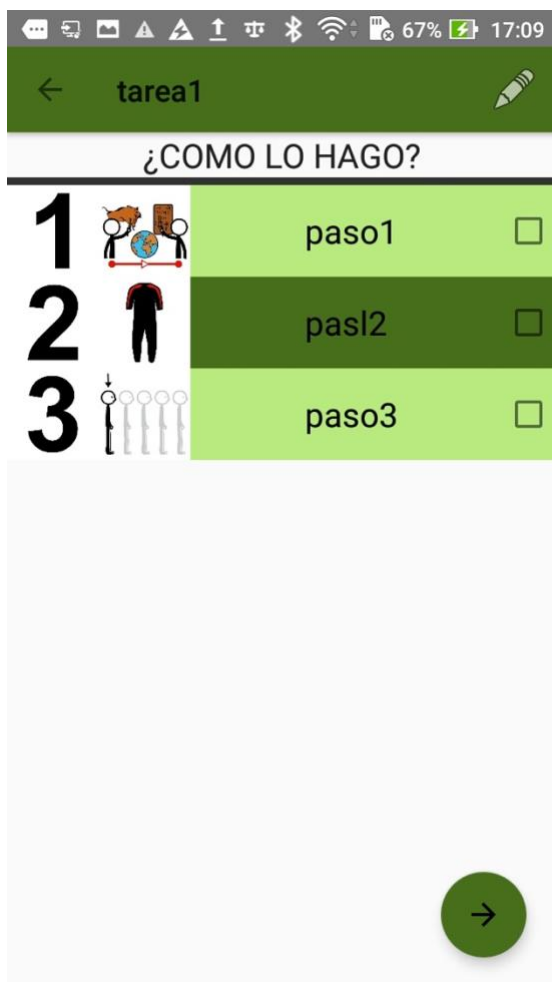
**Ilustración 0-11 Paso 2
Edición Tarea (Sociedad)**



**Ilustración 0-12 Paso 2
Edición Tarea (Objetos)**



**Ilustración 0-13 Paso 2
Edición Tarea (Tiempo)**



**Ilustración 0-15 Paso 3
Realizar Tarea**

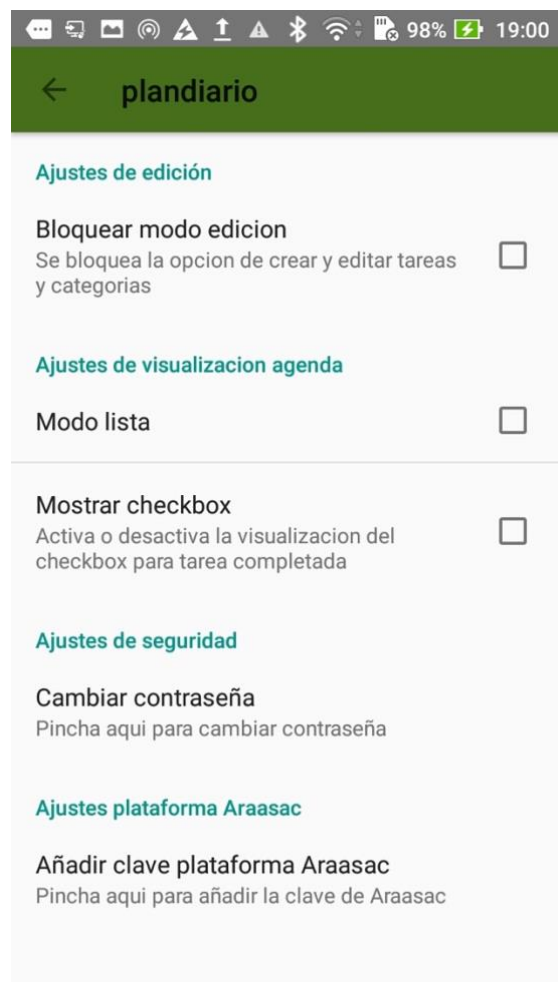


Ilustración 0-14 Ajustes